IUGN 16

Interak User Group Newsletter No.16 June 1987

CONTENTS

DATA PROTECTION ACT 1985:
Details of the members of the IUGN are held on computer file. Each member may view, alter or destroy any data held on him/her within that file. To obtain a copy of your file please send a stamped addressed envelope to the Editor. Your subsequent wishes regarding that data will be honoured without question.

NOTES

The CPMUGUK has a new bulletin board. Try it. 0753-868196. 300/300 & 1200/75. 8 bits, 1 stop, No parity full duplex. Run by Peter Catley, 39 Gordon Road, Windsor, Berkshire, SL4 3RG. This is a local call for 01 numbers.

Please note that the Greenbank Electronics "FREE" board repair service only applies to tracked boards. Nobody can be expected to repair sombody else's hand wired card.
If you adopt a sensible tactic to the construction of hand wired boards you can be sure of success.

1. Build the address decode - Test it.
2. Build a small part of the circuit. - Test it
   Continue until it all works.

Don't build it all at once, a little bit of construction and a test is the way to succeed.

Hmmm....
Ten people arive at a late night drinking club deep in Soho. At the end of the evening they had drunk 100 bottles of drink at a total cost of 100 pounds sterling. The empty bottles consisted of three types. An expensive Champagne costing 10 pounds a bottle, a wine costing 3 pounds a bottle and a small beer costing 50 pence a bottle. How many bottles of each type were drunk by the group.?

A friend recently let me borrow a book on CP/M that he had obtained to further his knowledge of this operating system. The book is called :-

MASTERING CP/M
by Alan R.Miller, published by SYBEX.

I found this book excellent. If you worked completly through the book you would finish up understanding both CP/M and Macro Assembler programming. Well recommended for members who have the basic idea of CP/M and wish to advance towards control over the system. Alan's descriptions and Macro library have added much to my enjoyment of the Interak. You will need MACRO-80.COM or some similar macro assembler to get full value from the book.

MINIMUM INTERAK - SPEC

MIB-3 ..... Z80A CPU card.
DRM-64 ..... 64k Ram card.
FDC-1 ..... Floppy disk controller card.
VDU-2K ..... Vdu controller card.
LKP-1 ..... Keyboard interface card.
ISBUS ..... Back-plane or mother-board.
PSU ..... Power supply.
DRIVES ..... 2 x 3.5" disk drives.
RACK ..... Mounting frame for above.
Keyboard ..... ASCII Keyboard.
CP/M ..... Disk operating system

SUGGESTED RACK LAYOUT

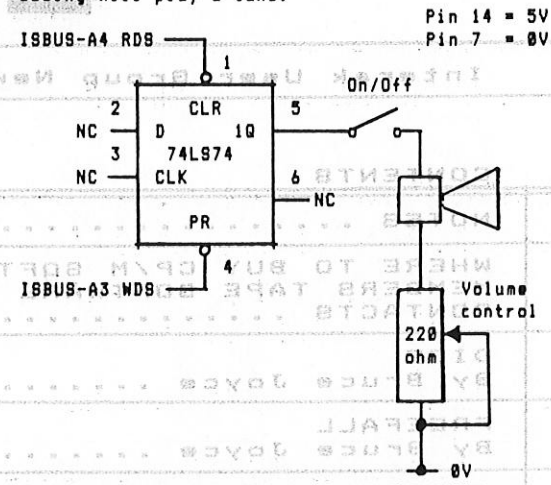
	1"	1"	1"	1"	1"	2"	2"	2"
← For Later Growth 6 slots spare.	VDU 2K	LKP 1 KEY BRD	DRAM 64 RAM	MZB 3 CPU	FDC 1 DSK CNT	DISK B	DISK A	PSU

I cut two positions away from the ISBUS to allow better access to the drives, and fitted all the remaining 11 edge connectors at once, this was to reduce the hassle of later expansion. Mount disks on a rigid card material, faced by a 2 inch front panel. They will then slide into the rack.

We have one lady member of the user group and a little bird has told me that her husband is now building his own machine, so as to get in on the act I suppose. I hope she will read this and perhaps write to give a womans view of the Interak. I wonder if she realises that I had to alter my mailing lables program when she joined, so that it could print either Mr or Ms. Up to then it had assumed Mr. : Very sexist.

A.S.A.M.
an
AUDIO SYSTEM ACTIVITY MONITOR

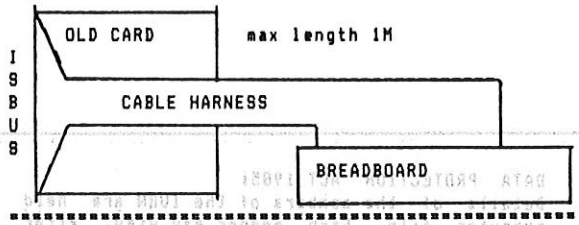
Connect a 74LS74 flip flop as shown. You can then hear your Interak running. In fact the right coding will play a tune.



The Read strobe must reset the flip flop.
The Write strobe must set the flip flop.
This ensures that current only flows through the loudspeaker from the start of a write instruction, and is cut off by the next op-code fetch. The volume control can be a pre-set or even a fixed resistor when you have obtained the desired volume level.
The on/off switch is optional, but I found it desirable in order to stop my cat going nuts when Dbase2 is sorting the user group mailing lists.
This little circuit is wide open to experimentation. You could add a divider chip and find the best sound division. If you do improve on it please let us all know.

A BREADBOARD INTERFACE

If you have an IC breadboard pad you can connect it to the Interack by getting an old unwanted card and connecting a cable harness to its edge connector. This allows the breadboard to be plugged into the computer for experimental purposes. You can put the patch circuit together on the breadboard and then link the required bus lines in as required, including 5V and 0V of course. You should keep the interface to about one meter so as not to induce to much noise on the bus.



Bob Eldridge

WHERE TO BUY CP/M SOFTWARE

CP/M v2.2 and v3.0 Disk operating system.
Greenbank Electronics, 460 New Chester road, Rock Ferry, Birkenhead, Merseyside, L42 2AE.
051-645-3391.

"C" language Compilers and Interpreters.
Grey Matter Ltd, 4 Prigg Meadow, Ashburton, Devon, TQ13 7DF.0364-53499.

ZORK 1,2 and 3. Sophisticated adventure games.
Anita Business Systems Ltd, London. 01-253-2444.

MEMBERS TAPE SOFTWARE - FOR SALE

You may use this section to sell tape software to other users. Send a brief description of your product giving details of its distribution and price, to the EDITOR. Note that you will be responsible for the support of your own product. See CONTACTS for "ORDER FROM" addresses. Software supplied is the responsibility of the "ORDER FROM". Please deal directly with the "ORDER FROM" in the event of bugs ect.

MACHINE CODE

NAME	DESCRIPTION	ORDER FROM:	COST
FIGFORTH	FORTH COMPILER	D.CAMBELL	£15.00
INTERPLAY	BB DRIVER	M&M ELECT	£ 4.00
MEGABUG	DEBUGGER	P.VELLA	£13.00
VELTEXT	TEXT EDITOR	P.VELLA	£ 5.00
XTAL BASIC	14K BASIC	P.VELLA	£40.00
ZYBASIC 3A	(ON TAPE)	GREENBANK	£15.95
ZYBASIC 3C	(IN ROM)	GREENBANK	£27.75
ZYMON 2.V203	MONITOR PROGRAM	GREENBANK	£15.95

CONTACTS

BACK ISSUES... Can be obtained from:-
D.Parkins, Greenbank Electronics, 460 New Chester road, Rock Ferry, Birkenhead, Merseyside, L42 2AE.

BOOKS..... Lend, borrow, and swop books via :-
R.E.Bowyer, 45 Ford drive, Yarnfield, Stone, Staffs.

D.CAMBELL .... 153 Lower Fairmead road, Yeovil, Somerset, BA21 5SR. Tel 0935-78282.

DISK LIBRARY . Public domain disk software from :-
Mr C.V.Bridgstock, 32 Wimborne ave, Thingwall, Wirral, Merseyside, L61 7UL. Phone 051-648-3000.

EDITOR..... Send submissions to :-
R.Eldridge, 28 Wycherley Close, Blackheath,London,SE3 7QH.

GREENBANK .... D.Parkins, Greenbank Electronics, 460 New Chester road, Rock Ferry, Birkenhead, Merseyside, L42 2AE.

M&M ELECT .... 8 Ayre view, Bride, Isle of man.

M.SAUNDERS ... M.Saunders, 7 Drumcliff road, Thurnby Lodge,Leicester,LE5 2LH.

MEMBERSHIP.... To join, renew or change your details contact :-
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.

P.VELLA ..... ~~10 Ford Drive, Yarnfield~~, Staffs.

SUBSCRIPTIONS. For information and payments please contact:-
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.

DICE

By Bruce Joyce
For ZYBASIC (disk version)

```
10 REM DICE BY BRUCE JOYCE
20 CLS
100 REM THROW DICE
105 FORX=6TO26: SET(X,42): SET(X,32): NEXTX:
    FORX=32TO42: SET(6,X): SET(26,X): NEXTX
106 FORX=1TO6
110 RESET(10,40): RESET(22,40): RESET(10,37):
    RESET(16,37): RESET(22,37): RESET(10,34):
    RESET(22,34)
120 D=RND(6)
130 IFD=1 SET(16,37)
140 IFD=2 SET(10,40): SET(22,34)
150 IFD=3 SET(10,40): SET(16,37): SET(22,34)
160 IFD=4 SET(10,40): SET(22,40): SET(10,34):
    SET(22,34)
170 IFD=5 SET(10,40): SET(22,40): SET(16,37):
    SET(10,34): SET(22,34)
180 IFD=6 SET(10,40): SET(22,40): SET(10,37):
    SET(22,37): SET(10,34): SET(22,34)
182 FORT=1TO50: NEXTT: NEXTX
183 ?AT(1,6)"PRESS"
185 INKEY X: IF X=0 GOTO185
187 ?AT(1,6)"     "
190 GOTO106
```

FREEFALL

By Bruce Joyce

For ZYBASIC (Disk version)

```
 1 REM*****************************************
 2 REM SET START LEVEL AND ZERO SCORE
 3 REM*****************************************
 5 L=1: S=0
 6 GOSUB400 : REM INTRO
 7 REM*****************************************
 8 REM SET VARIABLES USED IN EACH LEVEL
 9 REM*****************************************
 10 CLS: DOFF: SCROLL: LINE1
 20 A=30
 30 C=1
 40 T=30
 45 REM*****************************************
 46 REM PRINT HARD SURFACE
 47 REM*****************************************
 50 GOSUB350
 95 REM*****************************************
 96 REM MAIN ROUTINE TO MOVE MAN AND RAISE GROUND
 97 REM*****************************************
 100 INK.X: X=X-48
 105 POKE£F0C1+A,£20 : REM REMOVE IMAGE
 110 IFX=4A=A-1
 120 IFX=6A=A+1
 130 POKE£F100+A,0,£1D,0 : REM PRINT MAN
 132 IFC<20 GOTO140
 133 IF(C<30)*(PEEK(£F140+A+1))=32GOTO240
 134 IFPEEK(£F140+A+1)=£2A GOTO300
 135 IFPEEK(£F140+A+1)=£23 GOTO220
 139 REM*****************************************
 140 REM RAISE TUNNEL
 141 REM*****************************************
 150 R=RND(3)
 160 IFR=1T=T-1
 170 IFR=2T=T+1
 180 IFT<2 T=2
 190 IFT>50 T=50
 192 IF C=60+(L*20) GOSUB360 :REM PRINT SOFT GROUND
 195 C=C+1: IFC>50+L*20 ?: GOTO100
 200 ?TAB(T)"£::::::::£ "
 210 GOTO100
 215 REM*****************************************
 216 REM FALIURE
 217 REM*****************************************
 220 CLS: LINE10: ?"YOU HIT THE SIDE OF THE WALL"
 230 ?"YOUR FINAL SCORE IS ",S: STOP
 240 CLS: ?"YOU HAVE MISSED THE TUNNEL.": GOTO230
 250 REM*****************************************
 255 REM SUCCESS
 260 REM*****************************************
 300 LINE8: PAGE:
     ?"WELL DONE.YOU HAVE HAD A SOFT LANDING.
 305 S=S+L*10
 310 ?"YOU HAVE A SCORE OF ",S
 320 ?: ?"NOW TRY AGAIN"
 330 L=L+1: FORX=1TO6000: NEXTX: GOTO10
 340 REM*****************************************
 342 REM SURFACES
 344 REM*****************************************
 350 ?"£££££££££££££££££££££££££££££££££££££
     £££££££££££££££££": RETURN
 360 ?"**********************************************
     ********************": RETURN
 400 REM ***********************************
 410 REM INTRODUCTION
 420 REM***********************************
 430 CLS: LINE5: PAGE: DOFF
 440 ?TAB(25)"FREE-FALL": ?TAB(25)"---------"
 450 LINE8: ?"THIS IS THE LATEST SPORT ON THE
     PLANET 'SNIKRAP'.
 460 ?"IT IS CONSIDERED SO DANGEROUS THAT NOW THE
     FEDERATION OF INTER- GALACTIC BUSYBODIES (FIB)
     HAVE FORCED ITS PLAY TO BE DONE"
 470 ?"USING PROXIE-BOTS.
 480 ?"YOU CAN USE YOUR TERMINAL TO GUIDE THE P/B
     THROUGH THE FREE-    FALL HOLE": ?"GUIDE IT
     WITH KEYS '4'< AND '6'>.
 490 ?"DON'T HIT THE SIDES OF THE FREE FALL HOLE OR
     YOU LOSE.          IF YOU GET THROUGH THEN YOU
     CAN TRY A LONGER TUNNEL.
 500 ?"PRESS 'SPACE-BAR' WHEN READY.
 510 INK.X: IFX=32RETURN
 520 LINE17: ?"PRESS '          ' WHEN READY.":
     LINE17: GOTO500
1000 LINE1: SCROLL
```

A SESSION WITH AIRTEL

By Bob Eldridge

[Here is a sample of my first session with AIRTEL. This is an excellent BB well worth calling up on 01-200-3439. Any added comments by me in this cut down interaction are marked by []. I have reduced the 29 pages to give you the maximum info in the smallest space. I logged on to Airtel using 1200/75 baud.]

```
W E L C O M E

T O

A I R T E L

V21 & V23 AUTO SPEED SELECT SERVICE
8-BITS, 1-STOP, NO PARITY
OPEN 24 HOURS, 365 DAYS

FIRST NAME? BOB;ELDRIDGE;
SEARCHING USER FILE ...
CALLING FROM (AREA, CITY)? LONDON
SYSOP WELCOMES BOB ELDRIDGE
CALLING FROM LONDON
IS THIS CORRECT? Y
TERMINAL WIDTH (10-132)? 80

<A>TRS-80 4      <B>IBM PC        <C>TRS-80 1/3
<D>APPLE+//E     <E>BBC           <F>ATARI
<G>VIC-20        <H>EPSON QX-10   <I>APPLE MAC
ENTER LETTER OF YOUR TERMINAL, <CR> IF NOT LISTED:
CAN YOUR TERMINAL PRINT LOWER CASE? Y
Does your terminal need Line Feeds? Y
How many Nulls (0-50)? 0

Please Enter a 1-8 character Password to be used
for future logons.  This password may have any
printable characters you wish.  Lower case is
considered different from upper case and imbedded
blanks are legal.  REMEMBER THIS PASSWORD.  You
will need it to log on again.

Your password? ***
You have read through message 0
Current last message is 6975
You are caller number 6756
You are authorized  30 mins this call
Searching Message Base ...
You have no personal messages waiting.

* * AIRTEL MAIN HALL  * *
<T>ime   <G>oodbye
You must read the house rules of
AIRTEL before you have further access
<H>ouse ..... Rules for Airtel

Command: H
HOUSE RULES
~~~~~~~~~~~
Thank you for taking the trouble to read the House
Rules.  There aren't very many of them, as I don't
want the system to be tyranical. SO
ABIDE BY THEM.....<Please>.
                                        Thanks - Sysop
1.  Do NOT leave insulting messages.....I might
    get offended.
2.  Leave Messages in the relevant area....This
    helps others find them.
3.  If you want to use a pseudonym, then leave
    your correct name when you register. <FANX>.
Finally if you experience any problems, (System
Ones) drop me a line when you log off.

Thanks - Sysop.
<ENTER>

* * AIRTEL MAIN HALL  * *
<T>ime   <G>oodbye
If you now REGISTER completely you will be given
immediate access to AIRTEL
<R>egister ..... With Sysop
```

```
Command: R
NEW USERS REGISTRATION
**********************
If you wish to use the full facilities of this
System please register by answering the following
questions.  Please use your 'REAL' name.  If you
must use a Pseudonym, leave me a message as to the
reason why.
What is your FIRST NAME?
BOB
What is your  LAST NAME?
ELDRIDGE
I have your name as:
BOB
ELDRIDGE
Is this correct (Y/N) ?Y
What is your Address?
Street  ? 28 WYCHERLEY CLOSE
Town? BLACKHEATH
District? LONDON
Postcode? SE37QH
If NOT UK
Country     ? UK
What is your TELEPHONE number ? 01-853-4206

I have your details as :
28 WYCHERLEY CLOSE
BLACKHEATH
LONDON
SE37QH
UK
01-853-4206

Is this correct (Y/N)?Y
What MAKE of  computer
are you using.........:INTERAK (CP/M 2.2)
INTERAK (CP/M 2.2)
Do you confirm (Y/N)?
Y

Thank you for registering on this board.
Access type Granted :- RESTRICTED.....
Upgrades  will  be  issued  after  a  probationary
period !
Type    <ENTER>
```

```
* * AIRTEL MAIN HALL  * *
<T>ime   <G>oodbye
RESTRICTED Access granted
FULL Access granted
<1>Play ..... Adventure Games
<$>Tiger's ..... Trivia
<O>pen ..... Public message(s)
<C>hat ..... To Sysop
<Z>Angela's ..... Delight
<F>etch ..... all messages
<A>uction ..... Sales & Wants
<U>tilities ..... area
<S>IG ..... Special Interest Groups
<M>ail ..... Personal electronic mail
<D>ownload ..... file(s)
<X>exit ..... To Dos
```

```
Command: T
Logged On Since 01:46:02
Elapsed Time Is 00:12:34
```

```
Command: O
* *  OPEN MESSAGE AREA  * *
<R>ead ... Open Messages
<S>can ... Open Messages
<P>ost ... Open Messages
<E>xit ... Main Hall
<T>ime ... Elapsed time
<X>pert ... Change User Level
<G>oodbye ... Terminate session
```

```
Command: S
* *  SPECIAL INTEREST GROUPS  * *
<E>xit   <T>ime   <G>oodbye   <X>pert
<1>TRS-80 ..... SIG
<2>APPLE ..... SIG
<3>ATARI ST .... SIG
<4>IBM-PC ..... SIG
<5>CP/M ..... SIG
<6>BBC ..... SIG
<7>MSX ..... SIG
<8>Amstrad ..... SIG
<A>dventureland  SIG
<M>ail ... Electronic Mail Section
```

```
Command: U
* *  AIRTEL'S LOCAL UTILITIES  * *
<L>ook ... Examine Userlog
<S>earch .. Userlog for Name or Area
<I>nfo ... System Information
<F>ormat ... Reconfigure terminal
<B>ulletins
<P>assword ... Change your access password
<E>xit ... Main Hall
<X>pert ... Change user level
<O>ther Bulletin Board Systems
<U>pload files to this System
<T>ime ... Elapsed Time
<G>oodbye ... Terminate session
```

```
Command: 5
CP/M Special Interest Group
---------------------------
<R>ead CP/M Messages
<S>can CP/M Messages
<L>eave a CP/M Message
<O>ther Special Interest Groups
<E>xit back to AIRTEL
<X>pert ... Change User Level
<G>oodbye ... Terminate session
<T>ime ... Elapsed Time
```

```
Command: F
Terminal Width (10-132)? 79
<A>TRS-80 4      <B>IBM PC        <C>TRS-80 1/3
<D>APPLE+//e     <E>BBC           <F>ATARI
<G>VIC-20        <H>EPSON QX-10   <I>Apple Mac
Enter letter of your terminal, <CR> if not listed:
Can your terminal print lower case? Y
Does your terminal need Line Feeds? Y
How many Nulls (0-50)? 0
```

```
Command: O
Other Bulletin boards
* * * * * * * * * * *
<E>xit   <G>oodbye  <T>ime
<1>London area boards
<2>Eastern area boards
<3>South Eastern area boards
<4>South Western area boards
<5>Midlands area boards
<6>Scotland area boards
<7>Wales area boards
<8>Northern area boards
<9>Miscellaneous boards
```

A few House Rules
==========

General comments
Everything on this system is to be 'legal, honest, decent and truthful'. Obscene or offensive messages will not be tolerated. Nor will 'hacking' information, or requests for it. Any potential hackers are warned that both BT and the Fraud Squad are now very active in trying to curb this sort of activity.

Due to the pressure on the message space, I am going to ask people to follow a few 'rules' with messages. These are not hard and fast and I certainly don't intend to make the system stiff and formal, but keeping things tidy takes up quite a bit of time.

Rules:
---

1) Please delete messages when you have read them unless they are of general interest, or you really need to read them again. (You are asked if you want to do this if a message is adressed to you).
2) Please put machine specific messages on the appropriate SIG board (eg BBC messages on BBC sig etc.).
3) Message to sysop section is not for general queries about download programs, where you can get a new modem, how to play SNUDGE on a BEEB Model C, etc. This type of query should go in the main message area or the appropriate SIG. Your hard pressed Sysop doesn't have the time or the knowledge to deal with these sorts of query. You'll get a quicker and better response from 'all'.

Messages may well be deleted after a month or so, even if not read, and if you don't call for a couple of months you may get deleted from the userlog, this is purely due to disk space restrictions and the time taken to log on if there are too many messages and users in the log.

You are reminded that this is a free system, funded entirely from my own pocket (with a little bit of sponsorship). You can't expect a commercial type of service from something that you don't have to pay for. Adrian.

[This list of the boards was received from AIRTEL. The Interak with a Com-i card is best suited to the 300 and 1200/75 boards and so the list has been edited down to those only. If you require others then pick them off AIRTEL's utility frame.]

London region

(V) means Veiwdata format.

AIRTEL Colindale	01-200 3439 Adrian POP	300,1200/75 24 hours
ALICE'S RESTAURANT Palmers-Green	01-882 7573 John LAMBERT	300,1200/75 24 hours
ASYLUM Greenwich	01-853 3965 Mad-Vax	300 24 hours
BRIXTON ITeC Brixton	01-735 6153 Not applicable	1200/75 (V) 24 hours
CPMUGUK Windsor	0753-868196 Peter Catley	300,1200/75 0800-2300
COMMUNITEL Kensal Green	01-968 7402 Igor THOMAS	1200/75 (V) 24 hours
DATAFLEX FIDO Ponders-End	01-443 7020 NOT KNOWN	300,1200/75 24 hours
DISTEL Norbury	01 679 1888 Commercial	300 24 HOURS
FORUM-80 LONDON Wembley	01 902 2546 Victor SALEH*	300 Evenings Ring and Ask
GNOME AT HOME Muswell Hill	01-888 8894 Micrognome	1200/75 (V) 24 hours
HACKNEY BB Hackney	01-985 3322 Not applicable	1200/75 (V) 24 hours
HBBS LONDON Earls-Court	01-373 8302 Andrew HAMMOND	300,1200/75 24 hours
HEALTH-DATA Hackney	01-986 4360 Dr. C.J. DOBBING	1200/75 (V) 24 hours
ITCU EXCHANGE & MART Kensal Green	01-960 4742 Igot Thomas	1200/75 (V) 24 hours
LIBERTEL Brixton	01-733 7730 NOT KNOWN	1200/75 (V) 24 hours
LONDON BB Golders Green	01-455 6607 Daniel WILDER	300,1200/75 24 hours
MALCOMM-SYSTEMS Golders Green	01-458 9704 D. ZARGEL & M. C	300,1200/75 24 hours
MBBS MITCHAM Mitcham	01-648 0018 Martin NEWHAM	300,1200/75 24 hours
MEGA-ANCHOVY Chiswick	01-747 4662 James TREGASKIS	300,1200/75 24 hours
METROTEL E-Molesey	01-941 4285 Graham HANKER	1200/75 (V) 24 hours
MG-NET Surbiton	01 399 2136 Peter GOLDMAN*	300 Sun: 5-10pm
MICROLIVE - BBC Ealing	01-579 2288 David HARRISON	300 24 HOURS
MUSICTEL Golders-Green	01-455 0843 James ECKHARDT	300,1200/75 24 hours
NBBS - WANSTEAD Wanstead	01-530 5589 Perry HEATH	300,1200/75 w/d 10pm-8am w/e 2pm-8am
NETWORK 23 Edgware	01-958 7098 Bob SCHIFFREN	1200/75 (V) 24 hours
OPTEL Hampstead	01-794 0655 Not applicable	1200/75 (V) 24 hours
OSI LIVES! Pinner	01-429 3047 RB Frank LEONHARDT	300 24 hours
OWLTEL Bloomsbury	01-927 5820 Dave ATHERTON	1200/75 (V) 24 hours
POLY FIDO Bloomsbury	01-580 1690 Graham HOBSON	300,1200/75 24 hours
PRESTEL 300 BAUD Croydon	01-680 8245 Not applicable	300 24 hours
PROMETHEUS Sidcup	01-300 7177 Barry SPENCER	1200/75 (V) 8pm on
TAECOMM-INTERAK Hayes	01-573 8822 Thomas EVANS	300,1200/75 W/d 7pm-7am Sun 24 hrs
TECHNO-LINE Cricklewood	01-450 9764 Commercial	1200/75 (V) 24 hours
TUG - LONDON Hendon	01-200 7577 Steve FROESCHKE	300 24 hours
TWILIGHTZONE Richmond	01-788 0884 Steve EDWARDS	300 24 hours
TYPENET Beckenham	01-658 6942 John FELTHAM	300 24 hours

WBBS - WIMBLEDON Wimbledon	01-542 3772 Bob TYLER	300 M: 7pm-2am Sa: 7pm-2am
WEST LONDON TBBS Uxbridge	0895 52685 Iain PHILLIPS	300,1200/75 24 hours

Eastern Region

ACORN SERVICE E: Cambridge	0223 243642 Commercial	1200/75 (V) 24 Hours
APPLECRACKERS E: Basildon	0268 778956 Mike JONES	300 24 hours
BABBS I (BASUG) E: Felixstowe	0394 276306 Tony GAME	300 24 Hours
BASILDON ITEC E: Basildon	0268 22177 Tony DWYER	1200/75 (V) 24 hours
BASILDON ITEC E: Basildon	0268 25122 Tony DWYER	300 24 hours
ESTELLE E: Harlow	0279 441188 Commercial	1200/75 (V) Office Hours
ESTELLE E: Harlow	0279 443511 Commercial	300 Office Hours
INFERNO E: Colchester	0206 41401 Richard SMITH	300 W/e 2200-2330
MAPTEL E: Southend	0702 552941 Commercial	300 24 Hours
NBBS - HORNCHURCH E: Hornchurch	04024 73041 Julian ANDERSON	300,1200/75 24 hours
PETE's PLACE RCPM E: Colchester	0206 862354 Peter SMITH	300 24 hours
RICBBS E: Basildon	0268 710637	300 M/d 5-10pm M/e 24 hours

South-East Region

APE-HOUSE SE: Gosport	0705 524805 Stephen COLE	300,1200/75 24 hours
BBS09 SE: Portsmouth	0705 736025 John DUNSTER	300 We & Sa: 7-10pm Sun 10am-10pm
BEDHAMPTON BBB SE: Havant	0705 476285 N & B PINNER	300 Fri-Sun 6pm-11pm
BLUELIPS II SE: Thanet	0843 32637 Leroy MACFARLAND	300 Mo-Th 6pm-8am W/e: 1am-4pm
CADILLAC SE: Wokingham	0734 78568 NOT KNOWN	1200/75 7-10pm
COMPULINK - 1 SE: Woking	0483 573337 Frank THORNLEY	300 24 hours
DENTAL MICROBOARD SE: Whitstable	0227 276162 Derek WATSON	1200/75 24 hours
INS - BRIGHTON SE: Brighton	0273 773971 Quentin North	1200/75 (V) M't-8am W/e Testing
LABBS SE: Godstone	0883 844164 Pip CORDREY*	300,1200/75 24 hours
NKABBS SE: Medway	0795 842324 RB Dave FROST*	300 930pm-M't
PBBS SE: Woking	04862 25174 Mike PARKER	300,1200/75 24 Hours
PD S'ARE LIB. SE: E-Grinstead	0342 315636 Rod SMITH	300 24 hours
RBOB SE: Potters Bar	0707 52242 Various	1200/75 (V) 24 hours
SANCTUARY SE: Egham	0784 38110 Barry BRUMITT	300 24 hours
SBBS - WATFORD SE: Watford	0923 676644 Simon TALBOT	300,1200/75 9pm-8am
SEMINARY SE: Maidstone	0892 436850440 Vic YOUNG	1200/75 (V) 24 hours
SWAFAX III SE: Maidstone	0622 858304 Vic YOUNG	1200/75 (V) 24 hours
THE PROMENADE SE: Margate	0843 23029 Jim BRETT	1200/75 Fri 6pm-Mon 8am
VULCAN BOARD SE: Marlow	06284 6691 Vulcan (Karim)	300,1200/75 24 hours
WORTHING FIDO SE: Worthing	0903 212552 NOT KNOWN	300,1200/75 24 hours

South-West Region

BABBS - BATH SW: Bath	0225 23276 RB Mark TEMPLEMAN	300 W/d 9pm-M't W/e 9pm-9am
CBBS - SW SW: Exeter	0392 53116 Boyd HITCHCOCK	300,1200/75 24 HOURS
COMPUTERS FOR CHRIST SW: Exmouth	0395 272611 J. Burden	300 24 hours
HYTEK SW: Crediton	03634 802 Commercial	1200/75 (V) 24 hours
NWOBBS SW: Calne	0249 815204 Ian ANON	300 6pm-M't
OCTOPUS SW: Bristol	0272 421196 Peter EVANS	300 M/d 6pm-830am M/e 24 hours
SWINTEL SW: Swindon	0793 38793 Tony BAUM	300 9pm-8-am
TBBS - BLANDFORD (OZ) SW: Blandford	0258 54494 Leo KNAGGS	300 24 Hours
TI 4A/BC IPS SW: Chard	04606 4511 Alan DAVEY	300 On request Sun 10am-10pm
WARPZONE SW: Bristol	0272 541878 John JAMES	300 W/d 9pm-8am W/e 2pm-M't

Midland Region

ACCESS FIDO M: Worcester	0905 52536 J W BROOKS	300 24 hours
AMRAD M: Solihull	021 744 1558 R. (G4PYR) MASKI	300 24 hours
BLABBI M: Kettering	0536 511155	300
BLOXHAM NOBB M: Banbury	0295 720812 Alex CRAWFORD	300 10pm-1am
BOONA GOONA ISLAND M: Stoke-on-Trent	0782 631469 Jarod NASH	300 1030pm-8am
CBABBS - BIRMINGHAM M: Birmingham	021 430 3761 Mick COLEMAN	300 24 hrs (Ex Th)
EARWIG BBS M: Nottingham	0602 274369 RB Mike JERVIS	300 W/d 9pm-7am W/e 24 hours
FORUM-80 SPA M: Leamington-Spa	0926 39871 M. J. RANDLE	300 2230-0030

MAILBOX-83 - WARLEY 0384 635336		300
M: West-Midlands	Jim RODEN	w/d 8pm-830am
		w/e 24 hours
MILLEWAYS	0533 608442	300
M: Leicester		w/d 6-10pm
		w/e 10am-10pm
NORVIEW	0604 20441	1200/75 (V)
M: Northampton	Not applicable	24 hours
STARBASE OMEGA	0533 710889	300,1200/75
M: Leicester	Peter RUSSELL	Fri 11pm+
STOKE ITeC	0782 265078	1200/75 (V)
M: Stoke-on-Trent	Not known	24 hours
TORTURE CHAMBER	021 445 1219	300,1200/75
M: Bromsgrove	Jonathan VALE	24 hours
TUG II	021 444 1484	300
M: Birmingham	P & S Franchi	24 hours

Wales

BRECON BBS	0874 730692	300
WA: Bwlch	P. WAGG	9pm-8am
CARDIFF ITeC TELED	0222 464725	1200/75 (V)
WA: Cardiff	Chris PARKER	24 hours
CYMRUTEL	0492 49194	1200/75 (V)
WA: Colwyn-Bay	Ian Woodroffe	24 hours
MGBBS	0443 733343	300,1200/75
WA: Maerdy	Paul SENGUPTA	24 hours
NBBS - SOUTH	0633 366467	300,1200/75
M: Cymbran	Steve JOHN	8pm-8am
OBBS - PENARTH	0222 704739	300
M: Penarth	Walt DAVIDSON	24 hours
TECHNOFRESH	0570 423082	300,1200/75
M: Lampeter	Llew DICKINSON	24 hours

Northern Region

BBB	061 740 9306	300
NW: Manchester	NOT KNOWN+	7pm-11pm
EAST CLEVELAND BB	0287 43920	300,1200/75
NE: Middlesborough	Steve DENT	
FORUM-80 HULL	0482 859169	300
NE: Hull	Fred BROWN	W/d 730-1130pm
		W/e M'y-1130pm
HAMNET	0482 497150	300
NE: Hull	Joe LAWRENCE	W/d 6pm-8am
		W/e 24 hours
HBBS£1	0274 45246	300,1200/75
NE: Bradford	Oscar BRUMWELL	5-11
HOLE IN THE WALL	0742 350319	300
NE: Sheffield	John LOCKWOOD	24 hours
KCC-RCS	0385 890353	300
NE: Chester-Le-Str	NOT KNOWN	7pm-8am
LEEDS BB	0532 667385	300
NE: Leeds	Mike BARRETT	w/d 10pm-7am
		w/e M'y-M't
LOG-ON-THE-TYNE FI	091 477 3339	300
NE: Newcastle	NOT KNOWN	10pm-7pm
MATRIX	061-736 8449	300,1200/75
NW: Manchester	Ken FARNEN	24 hours
MBBS - LECONFIELD	0401 50745	300
NE: Beverley	MARTIN TAYLOR	24 hours
MBBS - MANTEL (OBB	061 748 9101	300
NW: Manchester	Lee FREESTONE	7-1030pm
MCIS	061 773 7739	300,1200/75
NW: Manchester	Mektronic	24 hours
MORECAMBE OBBS	0524 426133	300,1200/75
NW: Morecambe	Roy HEALD	24 hours
NBBS - CHESHIRE	0270 767025	300,1200/75
NW: Sandbach	David JACKSON	24 hours
NBBS - DERBYSHIRE	0246 865843	300,1200/75
NE: Chesterfield	Andrew SWAIN	7-11pm
NBBS - NORTHWEST	0695 421493	300,1200/75
NW: Aughton Green	G.R. OLDHAMS	24 hours
OBBS 1	061 427 1596	300,1200/75
NW: Manchester	Robert O'DONNELL	24 hours
OBBS GRIMSBY	0472 47896	300,1200/75
NE: Grimsby	Graham CHESTER	
OBBS TELEMAC 15	0625 33703	300,1200/75
NW: Macclesfield	Andy HOPWOOD	24 hours
ON-LINE SYSTEM	0429 34346	1200/75 (V)
NE:Cleveland	John HUNTER	24 hours
PBBS - NI	076162 RB	300,1200/75
NI: Bangor	J. Stringer	9pm-9am
SCITEL	0724 854365	1200/75 (V)
NE: Scunthorpe	Steve OLDFIELD	W/d 1-4pm
SHARON BB	0765 5296	300
NE: Ripon	Paul Richard GIL	Complicated
TARDIS	0772 27236	300
NW: Preston	DAVIES and CUNLI	W/d 630pm-9am
		Su: 24 hours
TIMEZONE BBS	0244 677978 RB	300
NW: Chester	Paul GAULTON	630-1130pm
		9am-1130pm

Miscellaneous - Regions

FBBS - JERSEY	0534 55855	300
CI: Jersey	Mike SURCOUF	6pm-6am
DUBBS - Dublin	0001 885634	300
EI: Dublin	Stephen KEARON	w/d 8pm-8am
		w/e 24 hours

*** Call Time Limit Reached...
*** TBBS Disconnected: 02:18:00

CHIPS

By F.R.Johnson

For DBASE2

(Available from the disk library IUG-7)

British Telecom,
Dept OM 44,
Telephone House,
Canterbury,
Kent,
CT1 3BA.

Dear David,

Please find enclosed a copy of the DBASE2 program which I use to catalogue my integrated circuits stock.

I started to develop it when I was still using the VDU2K card and later modified it to suit my Televideo 950 VDU. You will see from the program that it will cater for both screen widths so it can be run by any Interak user. Im sorry it's so big, (again), but I keep adding new facilities and "mug-traps". I also like to put in plenty of comments so that I don't forget how it works! I only use it at present for IC's, but I shall probally include all my other semiconductor stocks later when I have the time to look up all the data. The program is modular so that with a few changes of headings and variables it could easily be modified to suit other uses.

*ED - To use this program you should construct the following CMD files as listed below. Also create a DBF and two NDX files as detailed at the end of the article.

C.CMD See listings below. DBASE entry point.
NUMBER.CMD See listings below
TYPE.CMD See listings below
ADD.CMD See listings below
DELETE.CMD See listings below
CHANGE.CMD See listings below
CHIPS.DBF Build details at end of this article.
NUM.NDX Build details at end of this article.
TYP.NDX Build details at end of this article.]

```
*************************************************
*  A PROGRAMME TO ACCESS THE "CHIPS" DATABASE   *
*           WRITTEN BY F.R.Johnson              *
* This program was configured to run on a 64    *
* column VDU2K card or a serial VDU with 80     *
* columns. This listing is therefore written    *
* for the minimum width. Use is made of the     *
* VDU2K's inverse video facility, (variables REV *
* and NORM), to highlight certain characters.   *
* This does not affect the programmers Televideo *
* VDU, but it should be born in mind if a       *
* different type is used.                       *
* The program is menu driven and,(hopefully),   *
* idiot proof!                                  *
*************************************************
************************** Configuration Section *
CLEAR
SET BELL OFF
SET TALK OFF
STORE CHR(27) + "3" TO REV
STORE CHR(27) + "4" TO NORM
STORE 0 TO DEL
STORE " " TO KEY
STORE 0 TO MATCH
STORE 0 TO FLAG1
STORE 14 TO COL
STORE " " TO CHOICE
STORE "          " TO TYP
STORE "N" TO PRINT
STORE "Number      " TO HEAD1A
STORE "Type" TO TB
STORE "       Description" TO HEAD1B
STORE "Number      Type        Description;
                   Pack  Bin  Quantity" TO HEAD3
STORE "-------------------------------------;
--------------------" TO UL1
STORE "-------------------------------------------";
 TO UL2
STORE "-------------------------------------;
-------------------------------------------" TO UL3
STORE "************** INTEGRATED CIRCUIT DATA;
 BASE **************" TO TOP
STORE "****************** Written by F.R.Johnson ;
******************" TO BOT
STORE "(Press RETURN for the main menu)" TO M
STORE "Output to the screen or the printer? "to SP
STORE "Press any key to continue " TO PRESS
STORE "Search complete, press 'R' to repeat or;
 'M' for main menu " TO SEARCH
STORE "                                          ;
                  " TO SPACE
STORE " " TO CHAR
STORE 0 TO DIFF
STORE 0 TO DI
STORE "." TO DOTS
STORE " " TO SPACES
STORE "Updating files - please wait" to UPDATE
STORE 1 TO R
STORE 3 TO C
********************************** File Section *
********************* Indexed on number and type *
USE CHIPS INDEX NUM,TYP
********************************** Main Section *
ERASE
************************************ Do border *
@ R,C SAY REV + TOP
DO WHILE R < 21
   STORE R + 1 TO R
   @ R,C SAY REV + "**"
   @ R,(C+56) SAY REV + "**"
ENDDO
@ R,C SAY REV + BOT
***************** How many characters on screen? *
DO WHILE CHAR <> "64" .AND. CHAR <> "80"
   STORE "64" TO CHAR
   * Default to 64 *
   @ 1,1 SAY NORM
   @ 11,13 SAY "Please enter the number of ;
characters"
   @ 12,13 SAY "on your screen  (64 or 80);
    " GET CHAR
   ?? CHR(7)
   READ
ENDDO
** Store the difference for screen layouts later *
STORE VAL(CHAR) - 64 TO DIFF
******************************** If 80 characters *
************ Add some to the various lines, etc. *
IF DIFF > 0
   STORE DIFF/2 TO DI
   STORE "********" + TOP + "********" TO TOP
   STORE "********" + BOT + "********" TO BOT
   STORE "................." TO DOTS
   STORE "         " TO SPACES
   STORE "----------------" + UL1 TO UL1
ENDIF
DO WHILE CHOICE <> "L"
************************************ Main menu *
   STORE 1 TO R
   STORE 3 TO C
   ERASE
   @ R,C SAY REV + TOP
   DO WHILE R < 21
      STORE R + 1 TO R
      @ R,C SAY REV + "**"
      @ R,(C+56+DIFF) SAY REV + "**"
   ENDDO
   @ R,C SAY REV + BOT
   STORE " " TO CHOICE
   @ 1,1 SAY NORM
   @ 3,28+DI SAY "MAIN MENU"
   @ 4,28+DI SAY "=========="
   @ 6,COL SAY REV+"N"+NORM+"umber search for ;
IC............."+DOTS+REV+"N"
   @ 8,COL SAY REV+"T"+NORM+"ype search for ;
IC..............."+DOTS+REV+"T"
   @ 10,COL SAY REV+"A"+NORM+"dd IC's to ;
stock................"+DOTS+REV+"A"
   @ 12,COL SAY REV+"D"+NORM+"elete IC's from ;
stock............"+DOTS+REV+"D"
   @ 14,COL SAY REV+"C"+NORM+"hange ;
data........................"+DOTS+REV+"C"
   @ 16,COL SAY REV+"L"+NORM+"eave ;
program....................."+DOTS+REV+"L"
   @ 19,COL SAY NORM
   @ 19,COL SAY "Enter ;
choice....................."+DOTS;
```

```
   GET CHOICE
   ?? CHR(7)
   READ
   ********************** Convert to upper case *
   STORE !(CHOICE) TO CHOICE
   ***************** Load appropiate command file *
   DO CASE
      CASE CHOICE = "N"
         *************** Search for IC by number *
         DO NUMBER
      CASE CHOICE = "T"
         ***************** Search for IC by type *
         DO TYPE
      CASE CHOICE = "A"
         ******************* Add IC's to database *
         DO ADD
      CASE CHOICE = "D"
         ************* Delete IC's from database *
         DO DELETE
      CASE CHOICE = "C"
         ****************** Change stored IC data *
         DO CHANGE
      CASE CHOICE = "L"
         ************************* Leave program *
         ERASE
         STORE 0 TO TIME
*        DO WHILE TIME < 100
*           STORE TIME+1 TO TIME
*           IF TIME=20 .OR. TIME=40 .OR. TIME=60;
 .OR. TIME=80
*              @ 10,(TIME)/4+15 SAY "BYE"+CHR(7)
*           ENDIF
*        ENDDO
         ERASE
   ENDCASE
ENDDO
CLEAR
***************************** End of command file *

****************** NUMBER.CMD ********************
* A COMMAND FILE TO SEARCH THE 'CHIPS' DATABASE  *
* BY NUMBER                                      *
**************************************************
USE CHIPS INDEX NUM
STORE "R" TO KEY
STORE 0 TO ALL
******************************** Main search loop *
DO WHILE KEY = "R"
   ERASE
   STORE "          " TO NUM1
   STORE "S" TO PRINT
********************************** Instructions *
   @ 1,20 SAY SPACES+REV+"SEARCH FOR IC BY NUMBER;
"+NORM
   @ 3,0 SAY SPACES + "The program will find all ;
IC's whose leading characters match"
   @ 4,0 SAY SPACES + "the input.   e.g. entering;
 '74' will find '741' and '74L8244N'"
   @ 6,0 SAY SPACES + "Enter 'ALL' to list the ;
entire database"
   @ 8,0 SAY SPACES + "The resulting list may be ;
sent to the screen or printer (S/P)"
   @ 9,0 SAY SPACES + "(The default is to the ;
screen)"
   @ 12,0 SAY SPACES + "Enter number of IC to be ;
found "GET NUM1
   ?? CHR(7)
   @ 14,0 SAY SPACES + M
   READ
   ****** Trim spaces from entry otherwise it     *
   ****** wont find it                            *
   STORE TRIM(!(NUM1)) TO NUM1
   ************************** Something entered *
   IF NUM1 <> " "
      @ 14,0 SAY SPACES + SP GET PRINT
      * Ask for Screen or Printer *
      READ
      STORE !(PRINT) TO PRINT
      IF PRINT <> "P"
         STORE "T" TO DISP
         @ 16,0 SAY SPACES + "Do wish to display ;
 TYPE or BIN? (T/B) " GET DISP
         READ
         IF DISP = "B"
            STORE "Bin " TO TB
         ELSE
            STORE "Type" TO TB
         ENDIF
      ENDIF
      ERASE
      GO TOP
      ************************ Display or print *
      IF PRINT = "P"
         ERASE
         @ 10,10 SAY "Printing data....."
         SET FORMAT TO PRINT
         EJECT
         @ 1,0 SAY HEAD3
         @ 2,0 SAY UL3
      ELSE
         @ 1,0 SAY HEAD1A + TB + HEAD1B
         @ 2,0 SAY UL1
      ENDIF
      STORE 3 TO ROW
      ***************************** Show all IC's *
      IF NUM1 = "ALL"
         GO TOP
         DO WHILE .NOT. EOF
            STORE 1 TO MATCH
            @ ROW,0 SAY NUMBER
            IF DISP = "T" .OR. PRINT = "P"
               @ ROW,12 SAY TYPE
            ELSE
               @ ROW,12 SAY BIN
            ENDIF
            @ ROW,24 SAY DESCRIP
            ************** Extra data if printing *
            IF PRINT = "P"
               @ ROW,65 SAY PACK
               @ ROW,71 SAY BIN
               @ ROW,78 SAY QUANTITY
            ENDIF
            STORE ROW + 1 TO ROW
            SKIP
            ********************* Paging routine *
            IF ROW>21 .AND. PRINT <> "P"
               STORE " " TO KEY
               STORE 3 TO ROW
               ?? CHR(7)
               @ 23,1 SAY SPACES + PRESS GET KEY
               READ
               ERASE
               @ 1,0 SAY HEAD1A + TB + HEAD1B
               @ 2,0 SAY UL1
            ENDIF
         ENDDO
      ELSE
         **** If entered number is part of        *
         **** NUMBER, list it                     *
         ***************** Routine similar to ALL *
         FIND &NUM1
         DO WHILE @(NUM1,NUMBER) > 0 .AND. .NOT. ;
EOF
            STORE 1 TO MATCH
            @ ROW,0 SAY NUMBER
            IF DISP = "T" .OR. PRINT ="P"
               @ ROW,12 SAY TYPE
            ELSE
               @ ROW,12 SAY BIN
            ENDIF
            @ ROW,24 SAY DESCRIP

            IF PRINT = "P"
               @ ROW,65 SAY PACK
               @ ROW,71 SAY BIN
               @ ROW,78 SAY QUANTITY
            ENDIF
            STORE ROW + 1 to ROW
            SKIP
            IF ROW>21 .AND. PRINT <> "P"
               STORE " " TO KEY
               STORE 3 TO ROW
               ?? CHR(7)
               @ 23,1 SAY SPACES + PRESS GET KEY
               READ
               ERASE
               @ 1,0 SAY HEAD1A + TB + HEAD1B
               @ 2,0 SAY UL1
            ENDIF
         ENDDO
      ENDIF
      ************ Formfeed and turn off printer *
      IF PRINT = "P"
         EJECT
```

```
      SET FORMAT TO SCREEN
   ENDIF
   ************************** No match found *
   IF MATCH<>1
      ?? CHR(7)
      @ 4,0 SAY "IC not found"
   ENDIF
   ************************** Search complete *
   STORE " " TO KEY
   ?? CHR(7)
   @ 23,0 SAY SPACES + SEARCH GET KEY
   READ
   STORE 0 TO MATCH
ELSE
   STORE "M" TO KEY
ENDIF
ENDDO
STORE "S" TO PRINT
STORE " " TO KEY
STORE "Type" TO TB
RETURN
************************** End of command file *

***************** TYPE.CMD ***********************
* A COMMAND FILE TO SEARCH THE 'CHIPS' DATABASE  *
* BY TYPE                                        *
**************************************************
STORE "R" TO KEY
USE CHIPS INDEX TYP
DO WHILE KEY = "R"
   STORE "S" TO PRINT
   ERASE
   STORE "           " TO TYP
   ********************************* Instructions *
   @ 1,21 SAY SPACES + REV + "SEARCH FOR IC BY ;
TYPE" + NORM
   @ 3,13 SAY SPACES + "The following types of IC;
 are stocked"
   @ 4,13 SAY SPACES + "-------------------------;
------------"
   @ 6,6 SAY SPACES + "ARRAY          AUDIO;
       CMOS           COMSUMER  "
   @ 8,6 SAY SPACES + "COMPARATOR     CONVERTER;
     INTERFACE     MICRO "
   @ 10,6 SAY SPACES + "OP AMP         OPTO;
         POWER          RAM  "
   @ 12,6 SAY SPACES + "TIMER          TTL;
          VIDEO"
   ?? CHR(7)
   @ 15,0 SAY SPACES + "Enter type, (the 1st few ;
letters are sufficient) " GET TYP
   @ 17,0 SAY SPACES + M
   READ
   STORE TRIM(!(TYP)) TO TYP
   ************************** Something entered *
   IF TYP <> " "
      @ 17,0 SAY SPACES + SP GET PRINT
      READ
      STORE !(PRINT) TO PRINT
      ERASE
      GO TOP
      ************************ Display or print *
      IF PRINT = "P"
         ERASE
         @ 10,10 SAY SPACES + "Printing data...."
         SET FORMAT TO PRINT
         EJECT
         @ 1,0 SAY HEAD3
         @ 2,0 SAY UL3
      ELSE
         @ 1,0 SAY HEAD1A + "Type" + HEAD1B
         @ 2,0 SAY UL1
      ENDIF
      STORE 3 TO ROW
      ********************* Find all of that type *
      FIND &TYP
      *********** If entered type is part of TYPE,
      *********** list it *
      DO WHILE @(TYP,TYPE) > 0 .AND. .NOT. EOF
         STORE 1 TO MATCH
         @ ROW,0 SAY NUMBER
         @ ROW,12 SAY TYPE
         @ ROW,24 SAY DESCRIP
         IF PRINT = "P"
            @ ROW,65 SAY PACK
            @ ROW,71 SAY BIN
            @ ROW,78 SAY QUANTITY
         ENDIF
         STORE ROW + 1 TO ROW
         SKIP
         IF ROW > 21 .AND. PRINT <> "P"
            STORE " " TO KEY
            STORE 3 TO ROW
            ?? CHR(7)
            @ 23,1 SAY SPACES + PRESS GET KEY
            READ
            STORE !(KEY) TO KEY
            ERASE
            @ 1,0 SAY HEAD1A + "Type" + HEAD1B
            @ 2,0 SAY UL1
         ENDIF
      ENDDO
      ************ Formfeed and turn off printer *
      IF PRINT = "P"
         EJECT
         SET FORMAT TO SCREEN
      ENDIF
      ************************** No match found *
      IF MATCH <> 1
         ?? CHR(7)
         @ 4,0 SAY "IC not found"
      ENDIF
      ************************** Search complete *
      STORE " " TO KEY
      ?? CHR(7)
      @ 23,0 SAY SPACES + SEARCH GET KEY
      READ
      STORE !(KEY) TO KEY
      STORE 0 TO MATCH
   ELSE
      STORE "M" TO KEY
   ENDIF
   STORE "S" TO PRINT
ENDDO
RETURN
************************** End of command file *

****************** ADD.CMD ***********************
*   A COMMAND FILE TO ADD IC'S TO THE 'CHIPS'    *
*   DATABASE                                     *
**************************************************
USE CHIPS INDEX NUM,TYP
STORE "C" TO KEY
STORE 0 TO DUP
STORE "Y" TO CONT
************* Keep adding IC's until 'N' pressed *
DO WHILE CONT = "Y"
   ERASE
   APPEND BLANK
   @ 1,20 SAY SPACES+REV+"ADD IC'S TO THE ;
DATABASE"+NORM
   @ 4,10 SAY "Enter details of new chips below"
   @ 5,0 SAY "(Enter 'R' in 'IC number' to return;
 to the main menu)"
   @ 8,10 SAY "IC Number    " GET NUMBER
   @ 10,10 SAY "Type         " GET TYPE
   @ 12,10 SAY "Package      " GET PACK
   @ 14,10 SAY "Bin number   " GET BIN
   @ 16,10 SAY "Description  " GET DESCRIP
   @ 18,10 SAY "Quantity     " GET QUANTITY
   ?? CHR(7)
   READ
   *********** Upper case only on indexed fields *
   REPLACE NUMBER WITH !(NUMBER)
   REPLACE TYPE WITH !(TYPE)
   ************************ Check for 0 quantity *
   DO WHILE QUANTITY = 0 .AND. NUMBER<>"R"
      @ 18,10 SAY "Quantity     " GET QUANTITY
      ?? CHR(7)
      READ
   ENDDO
   ************************* Return to main menu *
   IF NUMBER = "R"
      ERASE
      @ 10,0 SAY UPDATE
      DELETE
      PACK
      RETURN
   ENDIF
   ******************* Check for existing record *
   STORE £ TO RECORD
   STORE NUMBER TO NUM1
   STORE " " TO KEY
   FIND &NUM1
```

```
   IF £ <> RECORD
      @ 20,0 SAY "IC exists please use CHANGE ;
- press any key " GET KEY
      ?? CHR(7)
      READ
      STORE " " TO KEY
      @ 20,0 SAY "                                ;
                 "
      STORE 1 TO DUP
      DELETE
   ENDIF
   ************************** More IC's to add? *
   STORE "Y" TO CONT
   @ 20,0 SAY "                                   ;
   @ 20,0 SAY "Add more IC's? (Y/N) " GET CONT
   ?? CHR(7)
   READ
   STORE !(CONT) TO CONT
ENDDO
********* End of main loop - return to main menu *
@ 23,0 SAY UPDATE
IF DUP = 1
   PACK
ENDIF
RETURN
***************************** End of command file *

***************** DELETE.CMD ***********************
* A COMMAND FILE TO DELETE IC'S FROM THE 'CHIPS' *
* DATABASE                                       *
**************************************************
USE CHIPS INDEX NUM,TYP
STORE "R" TO KEY
STORE 0 TO DEL
********** Stay in delete mode until 'M' pressed *
DO WHILE KEY = "R"
   ERASE
   STORE "           " TO NUM2
   @ 1,18 SAY SPACE8+REV+"DELETE AN IC FROM THE ;
DATABASE"+NORM
   @ 5,0 SAY "Enter the EXACT number of the IC to;
 be deleted"GET NUM2
   ?? CHR(7)
   READ
   GO TOP
   ***************************** Find entered IC *
   FIND &NUM2
   ERASE
   IF NUM2 <> NUMBER
      STORE " " TO KEY
      ?? CHR(7)
      @ 1,0 SAY "IC not found - press any key " ;
GET KEY
      READ
      STORE " " TO KEY
   ELSE
      @ 1,0 SAY HEAD1A + "Type" + HEAD1B
      @ 2,0 SAY UL1
      @ 3,0 SAY NUMBER
      @ 3,12 SAY TYPE
      @ 3,24 SAY DESCRIP
      STORE " " TO KEY
      ************************ Correct IC found? *
      STORE "Y" TO KEY
      @ 5,0 SAY "Is this the IC(s) to delete? ;
(Y/N) "GET KEY
      ?? CHR(7)
      READ
      STORE !(KEY) TO KEY
      ***************** Confirm IC to be deleted *
      IF KEY = "Y"
         DELETE
         ?? CHR(7)
         @ 7,0 SAY "Marked for deletion, please ;
confirm (Y/N) " GET KEY
         READ
         STORE !(KEY) TO KEY
      ENDIF
      ********************** Any more to delete? *
      IF KEY = "Y"
         STORE 1 TO DEL
         STORE " " TO KEY
         @ 9,0 SAY "Press 'R' to delete more IC's;
 or 'M' for main menu" GET KEY
         ?? CHR(7)
         READ
         STORE !(KEY) TO KEY
      ELSE
         ********************* Deletion cancelled *
         RECALL
         STORE " " TO KEY
         @ 9,0 SAY "Deletion cancelled, press 'R';
 to repeat or 'M' for main menu " GET KEY
         ?? CHR(7)
         READ
         STORE !(KEY) TO KEY
      ENDIF
   ENDIF
   ************************ Pack the deleted IC's *
   IF DEL = 1 .AND. KEY <> "R"
      ERASE
      @ 10,10 SAY UPDATE
      STORE 0 TO DEL
      PACK
   ENDIF
ENDDO
RETURN
***************************** End of command file *

****************** CHANGE.CMD ********************
* COMMAND FILE TO CHANGE DATA HELD ON THE 'CHIPS'*
* DATABASE                                       *
**************************************************
USE CHIPS INDEX NUM,TYP
STORE "R" TO KEY
STORE 1 TO ZERO
STORE 0 TO MATCH
********** Stay in change mode until 'M' pressed *
DO WHILE KEY = "R"
   ERASE
   STORE "           " TO NUM2
   @ 1,25 SAY SPACE8 + REV + "MODIFY IC DATA" ;
+ NORM
   @ 5,0 SAY "Enter the EXACT number of the IC ";
 GET NUM2
   ?? CHR(7)
   READ
   GO TOP
   ******************** Find the IC to be changed *
   FIND &NUM2
   ERASE
   IF NUM2 <> NUMBER
      STORE " " TO KEY
      ?? CHR(7)
      @ 10,0 SAY "IC not found - press any key ";
 GET KEY
      READ
      STORE " " TO KEY
   ELSE
      STORE 1 TO MATCH
      ************ If found display the held data *
      @ 1,0 SAY "You may change any of the fields;
, press C/R if field correct"
      @ 4,10 SAY "IC Number   " GET NUMBER
      @ 6,10 SAY "Type        " GET TYPE
      @ 8,10 SAY "Package     " GET PACK
      @ 10,10 SAY "Bin number  " GET BIN
      @ 12,10 SAY "Description " GET DESCRIP
      @ 14,10 SAY "Quantity    " GET QUANTITY
      ?? CHR(7)
      READ
      REPLACE NUMBER WITH !(NUMBER)
      REPLACE TYPE WITH !(TYPE)
   ENDIF
   ****************** Check for zero in quantity *
   DO WHILE ZERO = 1 .AND MATCH = 1
      IF QUANTITY = 0
         STORE " " TO KEY
         ?? CHR(7)
         @ 17,0 SAY "You have entered '0' for ;
quantity, is this correct? (Y/N)" GET KEY
         READ
         STORE !(KEY) TO KEY
         ** 0 quantity is correct - delete entry *
         IF KEY = "Y"
            @ 19,0 SAY "Deleting entry, please ;
wait"
            DELETE
            PACK
            STORE 0 TO ZERO
         ELSE
         **** 0 quantity incorrect - enter again *
            @ 17,0 SAY SPACE
            @ 14,10 SAY "Quantity    " GET QUANTITY
```

```
         READ
         IF QUANTITY <> 0
            STORE 0 TO ZERO
         ENDIF
      ENDIF
   ELSE
      STORE 0 TO ZERO
   ENDIF
ENDDO
STORE 0 TO ZERO
STORE 1 TO ZERO
STORE " " TO KEY
ERASE
************************* Any more to change? *
@ 10,0 SAY "Press 'R' to repeat or 'M' for
main menu "GET KEY
?? CHR(7)
READ
STORE !(KEY) TO KEY
ENDDO
ERASE
***** Update the files if IC was found & changed *
IF MATCH = 1
   @ 10,0 SAY UPDATE
ENDIF
STORE 0 TO MATCH
RETURN
***************************** End of command file *
```

[Ed - Notes on additional files needed.]

It is required that you create a CHIPS.DBF file. Use the following :-

```
CREATE CHIPS
```

Using the following structure :-

FIELD	NAME	TYPE	WIDTH	DEC
001	NUMBER	C	010	0
002	TYPE	C	010	0
003	PACK	C	005	0
004	BIN	C	004	0
005	DESCRIP	C	040	0
006	QUANTITY	N	002	0

Put some data into CHIPS.DBF, something like the example below, except using your own stock details. (Your descriptions can of course be 40 characters long.)

NUMBER	TYPE	PACK	BIN	DESCRIP	QTY
4532	CMOS	DIL	C11	8-BIT PRI ENCODER	1
4585	CMOS	DIL	C11	4-BIT MAG COMPARITOR	3
555	TIMER	DIL	T10	TIMING CIRCUITS	14
556	TIMER	DIL	T10	DUAL 55 TIMER	2
709C	OP AMP	TO5	A1	HIGH PERFORMANCE	2
709P	OP AMP	DIL	A1	GENERAL PURPOSE	1
7486A	TTL	DIL	T1	QUAD 2-I/P EXOR GATES	3
7489	RAM	DIL	R2	64 BIT RAM	2
74LS42	TTL	DIL	T1	BCD-DECIMAL DECODER	1
74LS624	AUDIO	DIL	A2	VCO	1
74S11N	TTL	DIL	T1	TRIPLE 3I/P AND GATES	2
75150	TTL	DIL	T1	16 LINE MULTIPLEXER	4
7555IPA	TIMER	DIL	T10	CMOS 555 TIMER	2
7660	POWER	DIL	P6	DC/DC +15 TO -15V MAX	1
7805	POWER	TO220	P6	5V 1A SERIES REG	2
8131N	TTL	DIL	T1	6 BIT COMPARATOR	1
81LS97N	TTL	DIL	T1	OCTAL BUFFER 3-STATE	1
CA3001	VIDEO	TO5	A4	VIDEO & WIDE BAND AMP	3
CA3046	ARRAY	DIL	A3	5 NPN SILICON TR'S	2
CA3140E	OP AMP	DIL	A1	MOS/FET BIPOLAR O/P	1
HM6116P-3	RAM	DIL	R2	2K X 8 STATIC RAM	1
LM301AH	OP AMP	TO5	A1	GENERAL PURPOSE	1
LM380N	AUDIO	DIL	A2	2W AMP	1
LM387N	AUDIO	DIL	A2	DUAL LOW NOISE PRE-AMP	1
LM3900N	OP AMP	DIL	A1	QUAD NORTON AMP	3
N5558V	OP AMP	DIL	A1	DUAL 741 TYPE	2
SL2560B	ARRAY	DIL	A3	?	2
SN76110N	AUDIO	DIL	A2	STEREO PRE-AMP	1
SN76131N	AUDIO	DIL	A2	STEREO PRE-AMP	2
SPX26	OPTO	DIL	T13	OPTO-ISOLATOR	4
TBA820M	AUDIO	DIL	A2	2W AMP (@12V/8 OHMS)	1
TDA 1010A	AUDIO	SIL	A2	6 WATT AMP	1
TL497ACN	POWER	DIL	P6	DC-DC CONVERTER 30V	1
ULN2002A	ARRAY	DIL	A3	7-STAGE DARL DRIVERS	2
UPC 566H	OP AMP	SIL	A1	GENERAL PURPOSE	2

Having done this you should index the CHIPS.DBF file to construct NUM.NDX and TYP.NDX using :-

```
USE CHIPS
INDEX ON NUMBER TO NUM
INDEX ON TYPE TO TYP
```

CHIPS.CMD should then be ready to use.

DBASE2 CHIPS or if like me you have re-named DBASE2 to be DO.COM then DO CHIPS will start it up.]

********** THE END **********

AUNTIE DAVIDS PAGE

Well nieces and nephews, Auntie David is pleased to tell you that her talents have at last been recognised. (By herself at any rate). Henceforth she is to be known formally as "Dame Auntie David". However she is as humble as she ever was and you can still address her as plain Auntie David, after the conventional bow or curtsy. (Just in case any snoopers from Burke's Peerage are about we'd better hasten to say that "Dame" here means "Dame" as in Christmas Pantomime rather than "Dame" as in Order of the British Empire etc).

Although the capriciousness of the IUGN printing presses is such that the date this reaches you can be widely different from the date on the cover, in theory at least it is June 1987. Children, let Auntie tell you what fun she and your other Greenbank Auntie, Auntie John, have been having. We have been privileged to attend a seminar on the "most exciting development of the decade" (according to Zilog): the Z80280. This is also known as the Z280, the final version of the Z800, which Auntie thinks was the "most frustrating development of the decade".

The Z800 was the chip that never was; the expected enhancement and development of the microprocessor we all know and love, the Z80. When it was the Intel 8080 versus the Zilog Z80, Zilog had the advantage, which they lost, by dillying and dallying (and my, my, children - Auntie is no stranger to dillying or dallying). Instead of Greenbank and Zilog (hoorah, hoorah) we ended up with IBM and Intel (spit, spit).

What Intel did right (commercially we mean, not technically or ethically) was produce a whole series of parts with exquisitely planned obsolescence built in. In numerical order (and approximately chronological order) the series was 8008, 8080, 8085, 8086, 8088, 80186, 80286, 80386. Zilog of course tried the same stunt: Z80, Z800, Z8000, Z80000; so why didn't it work for them? There were two reasons: (1) the Z800 never appeared, and (2) the the Z8000, Z80000, had an entirely different architecture from the Z80. The strength of the Intel line (commercially at least) is that each later CPU is pretty well derived from the one which went before - to such an extent that it is possible to run programs written for the 8008 and 8080 on the 80386 (the actual machine code is different so the programs must be reassembled, but it does show how basically similar the whole range is).

What Intel realised, and what Zilog temporarily forgot, is that people don't want to chuck all they've learned away every two years (the current frequency at which new MPUs are developed). It is in the chip manufacturers' interest to encourage you to replace your silicon as regularly as possible, but the wise ones know they can only persuade you to do this if they will let you keep your skill and experience and transfer it to the new chip.

Zilog eventually came to their senses: they can only persuade us to throw away our Z80s if they give us something better, but which still lets us build on our Z80 experience. The basic appeal of the Z80 has never been in question (after all it is a superior chip to the 8080 which is the foundation of Intel's highly successful range). What we wanted was modern levels of integration and sophistication but still based on the Z80 architecture. We were promised it and disappointed when the Z800 was abandoned, but just in the nick of time a new promise has been made and fulfilled. (The Z80280 really does exist - Auntie bought as many as she could of the first production to come over from the USA, and if you write to her nicely, on a bank cheque, she may share some of her hoard with you.)

This is what we learned about the Z80280 at the seminar: The Z80280 has about 150,000 transistors in it (compared with about 40,000 in the Z80). This means the complexity is about equal to that of the Intel (spit, spit) 80286. The clock for the Z80280 runs at 20 Megahertz or so at present, and Zilog anticipate that as improvements in production are made they may be able to get up to 50 Megahertz. This is believable because the original 2.5 MHz Z80 design eventually achieved 8 Megahertz at its maturity. By considering the present 20 MHz Z80280 in its most favourable light processing speeds of 4 MIPS (4 million instructions per second) are possible. Nevertheless Zilog have not been silly in providing performance which cannot be used; the Z80280 has many features which allow it to be used with "ordinary" speed memories, and peripheral chips. A built in cache allows internal processing to take place at the speed of light (well the speed of electricity actually) only reducing speed momentarily when the cache needs more data.

The Z80280 works pretty well as a Z80 at switch on, (unlike the "almost Z80280", the Hitachi 64180, more of this another time perhaps) and recognises all of the existing Z80 op-codes. But when you want to fly, you can warp into an expanded dimension and use extra more powerful instructions which have been provided for those programmers with their eyes on the stars. Extra 16-bit instructions have been added, such as multiply and divide; and memory management instructions, in conjunction with an on chip memory management unit, allow an address space of 16 Megabytes. This is ideal for Interak, which has a defined 24-bit address bus (AB0-AB15 on side A and AB16-AB23 on side B).

On chip peripherals include: the cache memory already mentioned, 4 DMA Channels, 3 16-bit counter/timers, and a UART. Dynamic RAM is supported by a built in 10-bit refresh address counter. Unlike in the Z80 the refresh rate can be varied, and refresh can be disabled altogether if it is not needed. A special multiprocessor mode is provided for those greedy pigs for whom just one CPU is insufficient. When running at full speed with (relatively) slow memories wait states may be required. On the Z80280 the presence and duration of wait states is selectable under software control. Trap-handling to distinguish "system" and "user" access will make it easy to implement secure multitasking operating systems.

The Z80280 comes in a 68-pin package (don't worry, Greenbank has sockets to suit) and can be arranged to power up in Z80 mode (24-bit address, 8-bit data, Z80 interface control signals) or in Zilog's own Z-Bus mode (24-bit address, 16-bit data, Z-Bus control signals).

Auntie's page is not supposed to be technical, so she shan't say here how to implement the chip, but she can reveal that a few of her favourite nephews (young Bobbie B and little Charlie B to name but two) already have their own Z80280s to play with after school. No doubt the pages of "IUGN" in due course will contain further eulogies on this chip and more technical discussions. Greenbank at time of writing, as well as cornering the market in the first Z80280 chips, has also a secret hoard of Z80280 Preliminary Technical Manuals. They are not cheap but each one is a more highly prized document than any old spies' memoirs. Pssst, want to learn the secrets of enhanced interrupt vectored mode?

And at last anyone who has been sneered at because his computer does not contain a marvellous 80286 or 80386 can reply "That old thing? - I have a Z80280, which is much newer than yours. (So there!)"

Goodbye from your Auntie. Don't do anything I wouldn't do! (In computing terms I mean; with the old bad back there's no scope for much else.)

FLOPPY DISK DATA TRANSFER

By Simon Haller

In systems where interrupts and DMA are not available, floppy disks present a problem to both the software and hardware designers. When using 8" disks in double density, the rate at which the data is transferred is 500Kbits/s or one byte every 16 micro-secs. This is not much time, even for a microprocessor at 4MHz.

My floppy disk interface curcuit appeared in IUGN 9 along with a short description of some software to do the data transfer. I have found that occaisionally this software is not fast enough and I get read or write errors from CP/M. Although there is 16 micr-secs between bytes, the actual time that a program has to detect that another byte is ready to be transferred is only 11.5 micro-secs (from the 2797 data sheet).

The program reads a status port to determine what is happening and therefore what to do next. It has three options: it must wait longer, i.e loop round and read the status port again, or a byte is ready to be transferred or all the data has been transferred and there is no more to do. In the worst case situation, the next byte will be ready just after the status port has been read. The program will not see this byte until it reads the status port again when it will do the required read or write. So, after reading the status port for the first time, it must be capable of deciding that no bytes are ready yet, looping back and reading the status port again, finding that a byte is now ready and doing the transfer all within the 11.5 micro-seconds.

The timing is most critical when writing data to the 2797 chip. The recommended Interak method for data transfer was given in IUGN 7 page 12.

```
     LD D,81H
     JR DRQ
DAT: DEFB 0EDH,0A2H ;INI or OUTI
DRQ: IN A,(DPOLL)   ;Read status port
     AND D          ;Set flags
     JP M,DAT       ;If DRQ get/give data
     JP Z,DRQ       ;Loop until INTRQ
```

So, if DRQ (Data Transfer reQuest) is asserted just after the read of the status port, the sequence of instructions is :-

```
     AND D          ; 4 clock cycles
     JP M,DAT       ;10
     JP Z,DRQ       ;10
     IN A,(DPOLL)   ;11
     AND D          ; 4
     JP M,DAT       ;10
     OUTI           ;16
```

The figure after each instruction is the number of 4MHz clock cycles taken by the Z80 to execute it. The figure of 16 for the OUTI instruction is not the correct one here though as the 11.5 micro-secs refers to the time from DRQ being asserted and the write pulse to the 2797. It is therefore right to exclude completely the time for the first read of the status port, as DRQ is asserted after this read, but the write pulse from the OUTI occurs about 11 clock cycles into the instruction. The total time between DRQ being asserted and the write pulse to the 2797, called the service time, is 60 clock cycles or 15 micro-secs.

The program described in IUGN 9 was :-

```
     LD DE,data address
     LD C,FDC_data_port
     LD H,program_address_high
     JR WRTST
     JR WREND           ;End of command
     NOP
     NOP
WRTST:IN A,(Status_port) ;Read status port
     LD L,A
     JP (HL)
     JR WREND           ;End of command
     NOP
     NOP
     EX DE,HL           ;Transfer data
     OUTI
     EX DE,HL
     JP WRTST
```

In the worst case, as above, the sequence of instructions is :-

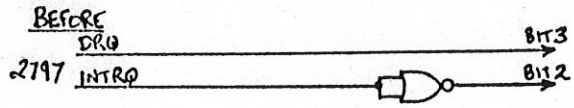
```
     LD L,A             ; 4 Clock cycles
     JP (HL)            ; 4
WRTST:IN A,(Status_port) ;11
     LD L,A             ; 4
     JP (HL)            ; 4
     EX DE,HL           ; 4
     OUTI               ;11
```

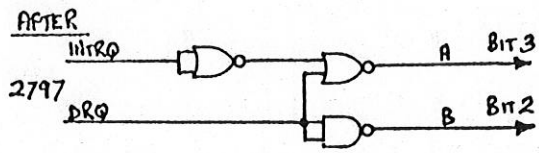
This gives a total of 42 cycles or 10.5 micro-secs which looks good. But what happens if the 2797 wants another byte immediately after it gets this one?

```
     OUTI               ; 5 Cycles (left over
                        ;   from previous write
     EX DE,HL           ; 4
     JP WRTST           ;10
WRTST:IN A,(Status_port) ;11
     LD L,A             ; 4
     JP (HL)            ; 4
     EX DE,HL           ; 4
     OUTI               ;11
```

These instructions take 53 cycles of 13.25 micro-secs which is too long and explains why I sometimes get errors.

To remedy this and reduce this time to below 11.5 micro-secs, a small hardware modification is necessary.





The extra two gates are spare on the board already so no new chips are needed. This logic performs the following transformation.

INTRQ	DRQ	A	B	
1	0	0	1	End of command
X	1	1	0	Data ready
0	0	1	1	Wait

The software is changed to :-

```
     LD DE,data address
     LD C,FDC_data_port
     LD H,program_address_high
     JR WRTST
     JR WREND           ;End of command
     NOP
     NOP
     EX DE,HL           ;Ready for data
     OUTI
     EX DE,HL
WRTST:IN A,(Status_port) ;Read status port
     LD L,A
     JP (HL)
```

The service time for the initial case has remained the same but for the second case, the sequence of instructions is now :-

```
      OUTI                ; 5 Cycles
      EX DE,HL            ; 4
WRTST:IN A,(Status_port)  ;11
      LD L,A              ; 4
      JP (HL)             ; 4
      EX DE,HL            ; 4
      OUTI                ;11
```

This gives a service time of 43 cycles or 10.75 micro-secs which is almost good enough. Since I have implimented this change, I have not had any errors due to timing.

========== THE END ==========

README.80

A Document file from Interak Disk library IUG-1

WELCOME TO FORTH

This is a public domain system, and may be freely distributed and copied, as long as the authors are given credit and no copyright notice is placed upon it. If we catch someone selling this system as their own proprietary product, with their copyright notice , we will do our best to make them regret it for the rest of their lives. You have been warned!!

This Forth uses the files of the host operating system. Although this reduces performance, it is much more portable and more convenient for novices. The use of files will be described later.

This disk contains some Huffman encoded files. They allow the system to fit on one floppy disk. Your first task is to expand them into the source files for the system. The Huffman files and their expansions are:

M80.HUF	-->	META80.BLK
K80.HUF	-->	KERNEL80.BLK
E80.HUF	-->	EXTEND80.BLK
C80.HUF	-->	CPU8080.BLK
UT.HUF	-->	UTILITY.BLK
HF.HUF	-->	HUFFMAN.BLK
CK.HUF	-->	CLOCK.BLK
FX.HUF	-->	F83-FIXS.TXT

In addition there are a few non-compressed files. These are the following:

README.80	This file which you are hopefully reading.
RUNME.COM	The file that you will run to create the BLK files.
EXPAND80.BLK	The file used to expand the HUF files.

In order to recover the actual source code, simply type RUNME and follow the directions. If all goes well, the corresponding BLK files will be created. You should of course make backup copies of the expanded source code as soon as it exists. We apologize for the time the expansion takes, but such is life. You are now in a position to modify and regenerate the system.

To do so, you should put META80.BLK and KERNEL80.BLK on drive B:, and put F83.COM on drive A:. Make sure there is room on A: for the new KERNEL.COM, then log onto drive B: and type the following:

```
B>A:F83 META80.BLK ( run F83 on META80.BLK )
OK ( loads meta compiler and generates KERNEL.COM )
BYE ( return to CP/M )
```

Now you have a new KERNEL.COM on A: and you are ready to add the selected extensions and make a new F83.COM (you did back up the old one didn't you?). Put EXTEND80.BLK, CPU8080.BLK, and UTILITY.BLK on A: with KERNEL.COM and type the following:

```
A>KERNEL EXTEND80.BLK (run KERNEL80 on EXTEND80.BLK)
OK  (loads all extensions and creates new version
     of F83.COM)
BYE (return to CP/M)
```

You can have a printout of these instructions by using Control P and entering TYPE README.80

In what follows the <CR> symbol represents the carriage return key on your terminal. To enter Forth from the CP/M environment, insert the disk containing the F83.COM file into drive A of your computer, (after booting it of course) and type

F83<CR>

Forth will reply with:

8080 Forth 83 Model 2.0.0
Modified 01Apr84

Forth is about ten years old now (in 1983) but some of the features of F83 are relatively new to this Public Domain Forth Model.

The VIEW command is one of the best. VIEW <word> <CR> will find the screen that contains the code for the <word>, provided that the file that contains <word> is on the currently logged disk drive. If you can fit all of the source code on a single drive, VIEW works great, and takes care of opening the file and displaying the screen for you. A recent addition to the system is the word FIX which combines the function of VIEW with that of the editor. To use it type FIX <word> <CR> and the screen upon which that word resides will be made available for editing, with the editing cursor placed following the first occurance of <word> on that screen.

We have also tried to make extensive use of the shadow screen concept. In this regard, the word A moves you from the current screen to its associated shadow screen. The SHADOW vocabulary contains a bunch of words that are helpful in maintaining shadow screens. COPY allows you to copy not only a screen, but also its associated shadow. SHOW allows you to list each source screen across from its shadow when you want to print them. Finally CONVEY allows you to move groups of screens and their associated shadows. To see what words are in the SHADOW vocabulary simply type SHADOW WORDS<CR>

There is also a decompiler present which reverses the compiling process, producing source code from object code. The user interface to it is the word SEE <name> where <name> is the name of a Forth word you wish to decompile. While the information SEE gives you is not as complete as that given by VIEW, a least it is always available, and not dependent upon whether or not the correct file is present on the currently logged disk drive. For a real experience try decompiling the words STATUS, which is a DEFERred word, ie an execution vector. SEE calls itself recursively in this case and tracks down the definition. This can be quite handy.

To see a list of the words that are in the dictionary type WORDS. You will see lots of words. Each is a command and each can be VIEWed. WORDS will stop if you hit any key. You can look at the word SHOW by typing VIEW SHOW. Assuming all is well you will see a Forth screen containing the definition of SHOW along with some other words. Type A L (cr) and you will be rewarded with the Shadow screen that goes with it. A L (cr) will Alternate List you back to the original screen of code. Look at it and when you see a word that doesn't seem fully self evident, VIEW it and read the shadow screen. The word SHOW and its source code that we have been viewing, is a command that prints out three source screens along with three shadows for a total of six screens per page. There are two versions of SHOW. The vocabulary you are in determines how SHOW will print out. If you are in the SHADOW vocabulary, six screens of code and shadows will print. If you are in FORTH 6 screens of code will be printed.

Now for some more commands. N is next screen and L is list. N L lists the next screen and B L lists the next screen back. N ED will edit the next screen etc. Now lets printout a few screens that will detail some of the editor commands. Most are from Starting Forth but some are new ones.

Type this: EDITOR VIEW JOIN<CR>

You are looking at a screen of editing commands. Lets go to the shadow with the A L command and do some exploring. While we are in the shadows lets look at the next five or six shadow screens that should be editor words. Use N L. If you have a printer you may print these screens now as follows. Type VIEW WIPE for example. If wipe is the first of 6 screens of editor words remember the screen number. If for instance it was 80 then type 80 85 SHADOW SHOW<CR>

Now a brief word about our mass storage interface. Our Forth 83 system runs as a guest under a host system's operating system. Because of this, we use the host's file system to contain our screen files. Screens are implemented as 1K blocks within a random access file. Screens are still treated as 16 lines of 64 characters when editing, with no embedded carriage returns or line feeds. In order to access a screen file it must first be opened. This can be done in two ways. The most common is to specify the name of the screen file on the execute line. Thus if you wanted to open MY.BLK you could fire up Forth with the following:

A>F83 MY.BLK<CR>

Once you are in Forth, you can open other files with the Forth word OPEN <name> where <name> is the name of the file you wish to open. OPEN is a defining word, and creates a dictionary entry using the file name. Once a file is opened BLOCK references are relative to the beginning of the currently open file. After a file has been opened with OPEN, it may be reopened by executing the file name. This is very fast, since only a pointer is changed in memory. One note, block 0 is still treated as a special case, and hence can't be loaded.

FORTH EXAMPLES

We would like to answer some of the most frequently asked questions about the F83 system at this point. The best way to gain familiarity with the system is to fire it up and try out the following examples. We will indicate the text that you type in as upper case text. The text in parentheses are our comments on what is going on. The <CR> indicates carriage return, as usual.

```
F83<CR> ( fire up forth from the CP/M environment)
8080 Forth 83 Model 2.0.0 ( Forth's reply and )
Modified 01Apr84          ( sign on message )
WORDS<CR>
empty mark hello ....
10 CREATE-FILE SAMPLE.BLK<CR> (Creates a file
                               called SAMPLE.BLK )
                              (which is 10 screens
                               big. )

1 LIST<CR>
Scr# 1
  0
  1
  2       ( This will be a blank screen )
  ...
 15

 ( Now we will edit screen 1 with the default dumb
   terminal editor which is pre-installed for you.
   Your first task is to install the cursor
   addressing routines to make the editor more
   convenient to use )

1 EDIT<CR>
Enter your id: .........
   ( Your id is a 10 character string that will
     automatically be placed in the upper right
     hand corner of line 0 of the current screen
     you are editing if you modify that screen I
     usually enter the date and my initials as
     follows: )
10MAR84HHL<CR>
   ( The screen will be blanked and listed, you
     should have a result similar to 1 LIST above,
     but with the line number also appearing on
```

```
    the right hand side of the screen )
   ( Now we will enter the source code necessary
     to install the cursor routine drivers for an
     ADM-3A terminal )
0 NEW<CR>
   ( This allows us to enter multiple lines of
     text.  The text input is terminated with a
     null line.  After each line is entered, it is
     redisplayed for you )
\ CURSOR ROUTINES FOR AN ADM-3A TERMINAL<CR>
   ( it is best to use line 0 for a comment )

EDITOR DEFINITIONS<CR>
   ( This will add the following definitions to
     the editor )
: ADM-AT<CR>
   27 EMIT ASCII = EMIT 32 + EMIT 32 + EMIT ;<CR>
: ADM-DARK<CR>
   CONTROL Z EMIT   ;<CR>
: ADM-3A<CR>
   ['] .ALL      IS  .SCREEN <CR>
   ['] ADM-AT    IS  AT <CR>
   ['] ADM-DARK  IS  DARK <CR>
   ['] NOOP      IS  -LINE <CR>
   ['] (BLOT)    IS  BLOT ; <CR>
ADM-3A <CR>
<CR>
DONE<CR>
1 modified
   ( Typing that last <CR> will return you to
     Forth.  The DONE command will write the text
     you entered to the disk, and exit from the
     editing environment.  To install the ADM-3A
     drivers you need now just type: )
1 LOAD<CR>
   ( Now the editor will work the way it was
     intended to with the current screen image
     always displayed at the top of the screen,
     and the line you are entering at the bottom
     of the screen.  To make sure type: )
1 EDIT<CR>
   ( The screen should be blanked, and the text
     you just entered will be displayed at the top
     of the screen.  The current line will be
     displayed at the bottom of the screen with a
     ^ pointing to the current editing cursor
     position.  The terminal's cursor will be
     below the line, and ready for an editing
     command. )
DONE<CR>
1 Unmodified
   ( Will again leave the editor and return to
     Forth  Now we will save the system and leave
     Forth )
SAVE-SYSTEM F.COM<CR>
   ( That will save the current system as a file
     called F.COM on the currently logged drive. )
BYE<CR>
   ( This exits Forth and returns to CP/M.  Now if
     you use F.COM instead of F83.COM you will
     have your terminal routines installed at boot
     up time. )
```

Note that the above routines are designed for an ADM-3A terminal, and will not work if your terminal is not an ADM- 3A. You will have to consult your terminal manual for the correct escape sequences required in order to position the cursor and implement the other special functions required. A better way to do the above is to recompile the entire system, with your terminal routines installed as the default instead of the DUMB terminal routines supplied with the system. In order to accomplish this you should copy the screen you just entered into the UTILITY.BLK file and recompile the system as described above. The following illustrates how to do this, using the multifile utility words implemented in F83.

```
F83 UTILITY.BLK<CR>
   ( files up F83 and opens UTILITY.BLK as the
     default file.  Make sure that UTILITY.BLK is
     on the currently logged drive.  Now find a
     blank or irelevant screen at the end of the
     editors set of screens, on top of which we
     are going to put the screen we just entered.
     Suppose it is screen number 32.  You should
     now type: )
FROM SAMPLE.BLK 1 32 COPY<CR>
   ( This will copy screen 1 from SAMPLE.BLK to
     screen 32 of the current file.  Note that
     SAMPLE.BLK must also be on the currently
     logged drive.  You can now recompile the high
     level portion of the by leaving Forth and
     typeing the following: )
BYE<CR>   ( leave Forth )
KERNEL EXTEND80.BLK<CR>
START<CR>
   ( Of course all of the necessary files should
     be present on the currently logged drive in
     order for this to work )
```

If you are new to Forth your best bet is to purchase the book STARTING FORTH by LEO BRODIE. It costs about seventeen dollars and is available at most large book stores. On page 84 you will find the commands for the line editor. Another valuable reference is the new book by MARTIN TRACY called FORTH TOOLS. It is available from MicroMotion, 12077 Wilshire Blvd. Suite 506, West Los Angeles, CA 90025. The price for the book is $20.00. The F83 model follows both the book and the Forth 83 standard.

The Forth Interest Group (FIG) has chapters in many areas. In the San Francisco Bay Area, meetings are held on the 4th Saturday of each month, currently at Chabot College in Hayward, CA. Call the FIG hot line if you wish to double check the location ph 962-8653 or the FIG Tree BBS (300 baud) at 538-3580.

Henry Laxen Mike Perry
1259 Cornell Avenue 1125 Bancroft Way
Berkeley, CA 94706 Berkeley, CA 94702

System Memory Map
The memory map will vary somewhat with CPU, and operating system, and options. For CP/M on the 8080 with 64K of memory, with four block buffers and all utilities, it is as follows: (all addresses in hexadecimal)

```
0100 Jump to cold start
0104 Jump to warm start
0108 ----

Dictionary with all utilities loaded.

5E75 ----HERE

Free space.

D10E ----SP0, TIB

| Text Input Buffer
V
^
Return Stack |

D1D6 ----RP0, >BUFFERS

Block Buffer Pointer Table

D200 ----FIRST

Block Buffers

E200 ----LIMIT
```

Standard System Documentation Requirement

1. The system dictionary space is CPU dependant and can be determined by typeing HERE U. <CR>

2. The application dictionary space is also CPU dependant and is the difference between the top of the dictionary and the beginning of the parameter stack. The location of the parameter stack varies depending on the amount of memory

available to the machine. The application dictionary space can be determined by typing SP@ HERE -U. <CR>

3. The data stack space is the same as the dictionary space.

4. The return stack space was arbitrarily set at 256 bytes. It can be altered by remote-compiling the system.

5. No mass storage block ranges are reserved by the system, other than the contents of the files that are distributed.

6. The user has available to him blocks 0 thru 65534. Note that block 0 may not be used for loading. Block number 65535 is reserved to indicate the buffer is missing.

7. Any ascii terminal should work with this system. If the user has a cursor addressable terminal, the editor can be easily modified to take advantage of the terminal's characteristics.

8. System action taken upon error conditions:

' <name> ['] <name> not found results in ? error message

*/ */MOD / /MOD MOD UM/MOD all division by 0 errors result in a 0 quotient

: in the case of an error, a ? error message will be printed

DO if a nesting error occurs, the system will crash. (if you are lucky)

EXECUTE if addr is not a compilation address, the system will crash. see DO

EXIT if the top of the return stack does not contain a valid return point, the system will crash. see DO

FORGET <name> if <name> is not found, a ? error message is printed. If the compilation vocabulary is forgotten, the system will crash. see DO

FORTH-83 if the error condition occurs, that this is not a standard system, I don't want to hear about it and I hope the system not only crashes, but burns.

LOAD if u is zero, the system will crash. see DO

==========

LISP.DOC
A document file from the Disk library IUG-1

LISP INTERPRETER APPLICATION NOTES

1. INTRODUCTION

The following application notes give some of the details necessary to properly operate the LISP interpreter which is implemented in the files named LISP.PAS/COM. As it has not been possible to address in the present context all of the features which would be required to master the subject, interested readers are invited to refer to the available litterature for further details. This author has found most useful the following textbooks :

Laurent SIKLOSSY - "Let's talk LISP" -
Prentice Hall, Inc., 1976

Patrick H.WINSTON - Berthold K.P.HORN - "LISP" -
Addison-Wensley Pub.Co., 1981

John ALLEN - "Anatomy of LISP" -
McGraw Hill Book Co., 1978.

The present interpreter has been derived from the LISP interpreter available in ZUG vol.£ 14. As in the original version, functions dealing with numbers are not supported.

2. HOW TO RUN LISP

Capital letters must be used for all LISP reserved words (essentially names of functions): because of this it is recommended that capital letters be used throughout.

To run the interpreter proceed as follows :

1. Type LISP followed by a <cr>.
2. Wait for the message R E A D Y to appear : this indicates that the interpreter is ready to evaluate a LISP expression.
3. Type the LISP expression to be evaluated on one or more successive lines (additional blanks do not produce any effect). The expression must be preceeded by a left parenthesis '(' and followed by a right parenthesis ')'. After having entered the LISP expression, type twice <cr> to initiate the evaluation. (Ensure that the correct number of parentheses is entered otherwise the interpreter will refuse to start the evaluation as it expects more input data).
4. Once the evaluation is completed, the interpreter will print out the resulting expression.
5. After that, other evaluation cycles can be introduced by repeating sequentially steps 3 and 4.

To terminate the process and return to CP/M type FIN followed by two <cr>'s.

3. INITLISP

INITLISP is the name of a file which must be available on the same diskette as the interpreter for the latter to work properly. Upon initiation of the interpreter, the contents of INITLISP is read and used to alter the LISP environment by adding user's defined variables (with their values) and/or functions.

INITLISP actually contains elements of a program written in LISP and any of the characters employed to write a LISP program may be used. However tabs (and perhaps other keys not representing alphanumeric characters, dots, commas, spaces, etc) should be avoided as they may produce unexpected results. Further, INITLISP must be terminated by the atom FIN followed by two <cr>'s to avoid reading beyond the eof.

INITLISP as presented only contains the atom FIN followed by two <cr>'s. In other words INITLISP will not affect at all the LISP environment upon activation of the interpreter.

The file INITLISP.STB, also included for the purpose of demonstration, contains a number of functions directly written in LISP. After renaming this file INITLISP (after having saved the original INITLISP for possible re-use), all these functions will become integral part of the LISP environment upon initiation of the interpreter. Hence they can be utilized as they were actual intrinsic function (with some loss of efficiency).

4. TERMINOLOGY

The following terms apply :

a. SEX

A LISP-EXPRESSION or SYMBOLIC-EXPRESSION (in short SEX) consists of an atom or a dotted-pair or a list (see further on).

b. ATOM

An ATOM consists of a string of up to IDLENGTH consecutive characters with the exclusion of '(', '.', ')', ' ', and <cr>. (IDLENGTH is equal to 10).

c. DOTTED-PAIR

A DOTTED-PAIR consists of a left parenthesis followed by an atom or another dotted-pair or a list, followed by a period, followed by an atom or a dotted-pair or a list, followed by a right parenthesis.

d. LIST

A LIST consists of a left parenthesis, followed by a sequence of elements (atoms or dotted-pairs or lists separated by one or more blanks), followed by a right parenthesis.

A list may contain any number of elements (zero included).

e. SPECIAL ATOMS

Atom T : denotes the truth value TRUE;
Atom NIL : denotes the truth value FALSE.

Atoms are always taken as representing variables unless they are preceeded by the atom QUOTE. In the first case the interpreter attempts to evaluate them (in other words looks for the value associated with them) while in the second case it does not do that.

5. INTRINSIC FUNCTION SUPPORTED

The list of the functions which are supported by the interpreter and a brief description of what each function does is given in the following. In all the examples which are given it is assumed that :-

the value of X is the list (A B C)
the value of Y is the list (D E F)
the value of Z is the atom K.

In other words it is assumed that the following three statements have been executed :

```
(SETQ X (QUOTE(A B C)))
(SETQ Y (QUOTE(D E F)))
(SETQ Z (QUOTE K))
```

Most of the material which follows has been taken from the text of Siklossy mentioned in the introduction.

AND
Number of arguments : any
Arguments : any SEXes
Value : NIL if the value of some argument is NIL, else T.
Example : (AND X Y) >>> T

APPEND
Number of arguments : 2
Arguments : lists, when evaluated
Value : the list of the SEXes representing the values of the two arguments
Example : (APPEND X Y) >>> (A B C D E F)

ATOM
Number of arguments : 1
Arguments : any SEX
Value : T if the arg is an atom; else NIL
Examples : (ATOM X) >>> NIL; (ATOM Z) >>> T

CAR
Number of arguments : 1
Argument : nonempty list, when evaluated
Value : first element of the list
Example : (CAR X) >>> A

CDR
Number of arguments : 1
Argument : nonempty list, when evaluated
Value : remaining part of list with first element deleted
Example : (CDR X) >>> (B C)

CONS
Number of arguments : 2
Arguments : first, any SEX; second, any list (when evaluated)
Value : a list, such that its CAR is the first argument and its CDR the second argument
Example : (CONS X Y) >>> ((A B C) D E F)

COPY
Number of arguments : 1
Argument : any SEX
Value : the argument
Effect : a copy of the argument is added to the environment.

DEFEXP (DEFine EXPression)
Number of arguments : 1
Use : to define a function and modify the environment accordingly; any such function always evaluates its arguments
Examples : refer to file INITLISP.STB

DEFFEXP (DEFine F-EXPression)
Number of arguments : 1
Use : same as DEFEXP, however the function does not evaluate its arguments

DEFMAC (DEFine MACro)
(NOT YET IMPLEMENTED)

EQ
Number of arguments : 2
Arguments : two SEXes
Value : T if the values of the 2 arguments are atoms with the same name or if they are lists contained in the same memory cells, otherwise NIL
Example : (EQ X Y) >>> NIL

EQUAL
Number of arguments : 2
Arguments : two SEXes
Value : T if the values of the 2 arguments are equal, else NIL
Example : (EQUAL X Y) >>> NIL

EVAL
Number of arguments : 1
Argument : any SEX
Value : the value of the value of the argument

LAST
Number of arguments : 1
Argument : a list, when evaluated
Value : the last SEX of the value of the argument
Example : (LAST X) >>> C

```
LENGTH
     Number of arguments : 1
     Argument : a list, when evaluated
     Value :  the number of SEXes at the top
              level of the value of the
              argument list
     Example : (LENGTH X) >>> 003

LIST
     Number of arguments : any
     Arguments : any SEX
     Value  :  a list of the values of the
               arguments
     Example: (LIST X Y) >>> ((A B C)(D E F))

NOT
     Number of arguments : 1
     Argument : any SEX
     Value    : T if SEX has value NIL,
                otherwise NIL
     Example  : (NOT X) >>> NIL

NULL
     Same as NOT

OR
     Number of arguments : any
     Arguments : any SEXes
     Value :  NIL if no argument value is non-NIL,
              otherwise T
     Example : (OR X Y) >>> NIL

PROG2
     Number of arguments : 2
     Arguments : any SEXes
     Value : value of the second argument
     Side effect : the arguments are
                   evaluated left to right

PROGN
     Number of arguments : any
     Arguments : any SEXes
     Value  :  value of the last (rightmost
               argument)
     Side effect : the arguments are
                   evaluated left to right

QUOTE
     Number of arguments : 1
     Argument : any SEX
     Value : the argument
     Example : (QUOTE X) >>> X

REMOB (REMove OBject)
     Number of arguments : 1
     Argument : an atom
     Value : NIL (but basically irrelevant)
     Effect : the atom is permanently removed
              from the environment

REPLACEH (REPLACE Head)
     Number of arguments : 2
     Arguments :  first : any SEX other than
                          an atom;
                  second : any SEX
     Value  :  the value of the first argument
               with its CAR replaced by the value
               of the second argument
     Side effect: the value of the first argument
                  is modified permanently
     Example: (REPLACEH X Y) >>> ((D E F) B C)

REPLACET (REPLACE Tail)
     Number of arguments : 2
     Arguments : as in REPLACEH
     Value :  the value of the first argument with
              its CDR replaced by the value of the
              second argument
     Side effect : as in REPLACEH
     Example : (REPLACET X Y) >>> (A D E F)

REVERSE
     Number of arguments : 1
     Argument : a list, when evaluated
     Value  :  the value of the argument with its
               elements reversed
     Example : (REVERSE X) >>> (C B A)

SET
     Number of arguments : 2
     Arguments : first : must evaluate to an atom;
                 second : any SEX
     Value : the value of the second argument; the
             value of the first argument has its
             value replaced by this value of SET
     Side effect :  the environment is permanently
                    modified

SETQ (SET Quote)
     Number of arguments : 2
     Arguments : first : an atom; second : any SEX
     Value  :  the value of the second argument;
               this value becomes the value of the
               first argument
     Side effect : the environment is permanently
                   modified

TRACE
     Number of arguments : 0
     Value : NIL (but basically irrelevant)
     Effect : all the functions will be traced

UNTRACE
     Number of arguments : 0
     Value : NIL (but basically irrelevant)
     Effect  :  the functions will no longer be
                traced
```

6. ADDITIONAL FEATURES

The following additional features are included :

PROG control structure, allowing iterative programming style (limitations : GO and RETURN cannot be used to jump outside PROG);

CONDitional expressions;

LABEL for temporary function definitions;

LAMBDA, FLAMBDA to introduce un-named functions evaluating (respectively non-evaluating) their arguments;

FUNCTION construct to deal with with the FUNARG problem.

For all of the above features it is recommended to refer to the available litterature on LISP.

7. ...TO TERMINATE

Although quite some testing has been performed it cannot be excluded that the interpreter still contains a substantial number of bugs. So I wish you good luck with it !

I will welcome any queries and/or suggestions in this or any other respect.

Lanfranco EMILIANI
Maurits de Brauwweg 11
2597 KD Den Haag
The Netherlands

Evans above

Tom Evans will be moving location very shortly. Please post all mail and Membership forms c/o GREENBANK ELECTRONICS at their usual address till further notice. Ta muchly Tom....

IBM KIT

The IBM compatible project is well underway now, with all cards collected and assembled, these consist of the Motherboard and Multi-function I/O card (both made by Eclipse, Taiwan), re-cycled genuine IBM colour garphics card, and an American Ompti hard disc controller, 5.25" floppy drive is also re-cycled from an old IBM. The 10Mb Hard drive is a Miniscribe from one of those fancy word-processors. The keyboard is throwing up some problems, this came from an Advance machine by Ferranti, and there appears to be a timing problem, and it refuses to send characters to the main machine at certain start-up times, I will try and get hold of a more compatible keyboard when funds allow, in the mean time I have found a way of "forcing" it to work. I took the lid off the keyboard to nose around but forgot about the sprung-loaded legs, of course these bounded out with a life of their own, and I spent the rest of the morning trying to figure out how they fitted back in the case, and "work" when fitted. Recently I managed to find another hard drive (Rodine 10Mb) in external case, so I decided to fit that on the spare controller slot, unfortunately both drives failed to work with each other, although when used separately both ran o.k? With no documentation it was going to be a hit or miss affair, but after a couple of days on and off experimenting, plus the loss of all data on the original hard drive when I tried vainly to access it, I fitted a dil switch to the jumper pad on the Miniscribe, and with great personal skill (my word for luck), I managed to work out the right combination, it turned out that the drive jumpers were set to hog all four select points, plus the ready signal was jammed on. It was lucky I had all my data on the vandalised drive backed up to floppies, mind you it took 20+ discs worth of data to replace the stuff I had lost (up to 360K per disc). I hate to think what will happen if I'm dumb enough to crash both drives. The only thing I don't like about the set-up, is the blast of wind that howls from the cooling fans, on a bad night its icicles on the "hooter" time, perhaps I should fit a heater element between the path of the blast, and my poor nose. I have just fitted a V20 replacement processor in place of the 8088, more about this cock-up next time!

Taecomm has now increased its user base to 500 at the time of writing, and gone "International" with calls from France, Copenhagen, and Australia coming in, plus a few more Interakers are giving it a go which cheers me up no end. There is a big problem arising through lack of available disc space, I can add another pair of drives to match the 8" double sided Tandons (if I can find a supplier at the right price), but this will only be solving the problem in the short term, what is really needed is a very large capacity Hard drive, these I can get at very reasonable (cheap) prices, but of course I will then run into the problems of interfacing the beasty. This problem will have to be sorted out soon as without extra storage it looks as though we will not be able to advance much further in the way of new features, extra Sig logs and upload/download files. Nothing has physically changed with hardware on Taecomm except that it is now connected together with an IBM PC via their respective RS232 ports, this allows me to write ascii files on the IBM when Interak is busy with running Taecomm, this also applies to database data, and when Interak is free, the respective files can be ported over ready for normal daily use. I have recently been using the IBM as Interak's terminal, this has been quite successful. I am using a comms program on the IBM configured to behave exactly like my ADDS terminal. With this set-up I have a full colour 80 column/graphic display dangling from the Interak, and the beauty of it is that I can leave the Interak doing its own thing, then jump out of the emulator, and run a program on the IBM at the same time. This gives ideas to make Interak into a powerful stand alone crunching machine, with a little Interak or IBM as a front end, that can act as a terminal, but still be able to sneak off and do little jobs on its own whilst leaving the monster busy crunching and burping huge amounts of data in the background. Perhaps a MK 2 Interak could be designed running as a parallel project to MK 1, with maybe using the larger style mini computer cards (VME, QBUS), and 32 bit cpu(s) plus something like a UNIX operating system. Maybe, maybe not.

LOCK, STOCK 'N' BARREL

What was supposed to be a lazy weekend at our retreat in Wales, took on a more hectic turn after I purchased a local paper at the village store. Taking a break and a large crusty bread sandwich, I thumbed through the local, and spotted the advert that did all the damage to our snoozy afternoon, the peace crashing words were "Most spacious period farmhouse with wealth of character", I read it out to Barbara and the boys, and within minutes we were on the road searching for this "Des Res", but we couldn't find it that afternoon, so resumed the search the following morning. We found it after about half hour, it turned out to be 4 miles from our existing place, we were fed wrong directions in the advert (crafty these estate agents). Anyway the driveway was over half a mile long, and when we made it into the yard we were greeted by the sight of a couple of naked children playing on rusting car wrecks which were scattered about the yard, the Dutch barn only had part of its roof still clinging onto the main structure, the depleted roof was flapping and creaking in the warm lazy breeze. We were shown round the place by the mother, followed closely by the children who just kept staring at us as if we were from another world. The house was quite large, with low beamed ceilings, and a great big walk-in fireplace, just the job on a cold winter night, to Barb's delight the kitchen ran the whole length of the house with mega-storage space, and a good sized pantry (tons of grub). Anyway while Barb was still looking round the house, the rest of us wandered down the track into the wood that surrounded the property, after about 5 minutes strolling along between the tall Oaks, the boys and I decided this was for us, so we turned back to link up with Barb, and after her tour of the house, she had also decided to take the plunge, and there it is, without unforseen cock-ups we should be installed within the next few months. As I say, if there is no complications the new address for membership renewals and applications will be given in the next issue (hopefully he says with crossed fingers, legs and other bits).

Sees yas....Tom

LETTERS

R.L.Ruddock,
46 Scholey Ave,
Woodsetts,
Nr. Worksop,
Notts.

Dear Sir,

I am writing to you in the hope that you can give me some help in finding an "Intelegraph" board. I believe you first mentioned this in IUGN 3+4.

About 2 years ago I enquired about this board from Computer Aids, Leicester, and was told that it was no longer available. Since then I've tried to find something similar, but had no luck.

I have a serial VDU board from A.M.Electronics, which is OK for general applications, but it has no graphics ability, so if you could maybe point me in the direction of a possible supplier, I'd be very grateful.

On the subject of past newsletters, many thanks for your series 'CP/M on the Interak'. It did a marvellous (and necessary) job of demolishing Digital Research's policy of using the longest possible words to describe the simplest possible operations.

Well done!

Yours
Rick.

[Ed- Thanks Rick, I am glad it was of some use. As to 80 column graphics screens I can only really think that Tom our secretary might give you the best help. He once coupled his Interak up to a colour graphics terminal card. Drop him a line to see if you can extract some ideas from him. The other possibility is to ask David of Greenbank about his developing design for an 80 column display. You may find that with his expert help you can construct a prototype on a DTI card until the production model becomes available.]

Scott Dadak, 1/3/1987
31 Pheonix St,
Bury,
Lancs,
BL9 0HS.

Dear ED,

I am donating the enclosed listing of "DEMOLITION". I have written for the Interak-1 system. All the necessary details are enclosed on the listing itself. Hope you and the other users enjoy playing the game as much as I do.

Yours Sincerely
Scott.

P.S. Please send a reply as I am not sure if my Dad is still in credit to receive the IUGN. If you have any ideas of what kind of programs are needed for publishing in the IUGN please enclose the details.

[Ed- I will be pleased to publish your excellent program in a later issue as I am short of space in the current ones. Most people ask for application programs. I.e programs that can do real work. But, if you enjoy writing good games please continue. I often think games programs can stretch the imagination of the programmer, which is real work in itself. One day the logic solutions that you find for your games will help you produce idea's that solve the bigger problems in the game of life. Very profound!!. Keep up the good work.]

17, Ranelagh Rd,
Sheerness,
Kent. ME12 2NS.

18th March 1987

Dear Sir,

About three years ago I began building an Interak 1 computer. Unfortuneately I ran out of time & money. I hope to be in a position shortly to resume construction.
Could you let me know if Interak is still available. If so, would you send me a current price list & details of any changes to the specifications.

Yours Faithfully

P. Apps

P.APPS

[Ed- What a picture this letter makes. Yes the Interak is still alive. David has sent you some updated information. The past three years has seen disks, a bigger screen and other improvements. I look forward to hearing from you again when "she" is on-line.]

John.D.Ritchie, 7/2/1987
S.T.E.F.
R.A.F. Finningley,
Nr Doncaster,
S.Yorks,
DN9 3LQ.

Dear Bob,

Profound apologies for not having written sooner, for not having returned your disc and for not having included anything on the disc, now that I have returned it.

Basically, since before Christmas, I have been run ragged. It is partly my own fault for wishing to be involved in everything that is going on; to wit two theatre groups (one service, one civvy,) Chairman of one committee, secretary of another, plus studying for HNC as well as being an active member of my local Mensa. As you can see breathing has to be scheduled in!

Oh I do look at the Interak, wistfully, and I do read the newsletters in my tea-breaks at work. Oh did I mention my full time job, plus duties of course.

Well it keeps me off the streets, and off the computer, of course. By the way I've purchased CP/M+ from the good David, though of course I haven't had time to run it yet!!

So much for the sob-story on to cheerier topics.

I got the two newsletters, the other day, as always packed full of good stuff. I really do appreciate the effort that goes into it, having been involved in a couple of station magazines, and I really do congratulate you on a job well done.

By the by I got the Chess Eprom from David free, gratis and for nothing. What a generous man he is. However, he can be forgetful, he sent me Zybasic 4 on disk a while ago, but no manual! I hate to pester him as I can imagine how busy he is, but I guess I will have too, as at the moment I can't figure out how to save programs and consequently reload them, and there are all the enhancements that I would take advantage of, if I knew them.

In conclusion, thank you again for all your kindness and I hope I may be able to do something in return one day.

Best regards
John

[Ed- Thank you for the letter and my returned disk. Sorry you are a little overloaded, the Interak will sit and wait until you have more time. As for contributions, don't worry, as I have more than I can handle at the moment. Some of the members work will take a full four issues to get printed and so you can relax on that front. Mind you next year could be a good time to produce something. You mentioned that you were involved in a magazine once, perhaps you could pass on some constructive idea's on how to make this newsletter more interesting. Looking forward to hearing from you in the future.]

Patrick Meehan, 9/2/1987
15 Pennine Road,
Chelmsford,
Essex.
CM1-2HG.

Dear Bob Eldridge,

I am writing this letter for the "Interaktion Newsletter" in responce to the leaflet that accompanied issue 14. I would like to take up the point that "Tom Evans" made about the same people contributing articles and receiving no feedback. My own reason and one which I think will apply to many people is that I do not have a working computer. It has taken me over one year to build five boards. The 'Interak' is not a inexpensive computer (not if it's built right). I wish to purchase an 'RS' switch mode P.S.U. for about 140 pounds, then there's a keyboard (a decent one like a processer controlled keyboard) is going to cost another 150-200 pounds. Therefore when articles are written about such and such a card or program, those of use who do not have working computers cannot comment.

The 'Interaktion users group' is still a very young organisation. Given time when members start getting their computers together, I'm sure far more feedback than at present will be received. When I started to build Interak I was on the dole and buying bits and pieces for the VDU-2K. It took eight weeks to save for the P.C.B. I even needed finacial assistance for that, like selling my very good HiFi (incuding my beautiful KEF speakers- WHAT A KILLER!).

As for the newsletter itself, I think its an exccellent compilation; and serves a very useful purpose. The article about the 'TAECOM' bulletin board in issue fourteen was very good and put a lot of anxiety about using a bulletin board to rest. Not having used a bulletin board before I was a little put off when reading the instructions in the 'TAECOM' newsletter.

Well thats about it, my first contribution to the IUGN, (not very good is it?). Bye for now.

Yours sincerely
Patrick Meehan.

[Ed- It is very good. You have clearly and honestly stated what is on your mind. People who build the Interak are clear headed thinkers and your contribution will be read and agreed with by many. As to your comments I do understand the cost problem. The Interak has the failing that it is initially an expensive computer to build. But, I know several people who, having bought and discarded three computers, are now saving money by not having to junk the Interak after a couple of years. Can I please correct you when you cost a switched mode P.S.U. Greenbank do one for just under 60 pounds which will drive a full rack and two 3.5" disk drives. Also a friend recently build a kit keyboard from 'MAPLIN' and I am certain it was much less that 200 pounds. When I built my machine I, like you, suffered the initial cost trauma. My solution was to build the minimum to get it to work then add the enhancements a bit at a time afterwards. Your comments regarding Tom's moans are accepted and when you finally get your machine up and running I will be happy to publish your work because, unlike those so called cheaper machines, we intend to still be here.]

Ian C.Vaudrey, 5/2/1987
23 Ramsey House,
Vassall Road,
SW19 6NB.

Dear Bob,

I'd like to offer some comments on your reply to the letter from B.A.V.Young, printed in IUGN 14 (with a few digressions).

Apart from my Interak, my other 'real' computer is a Tangerine, originally 6502 based, but for the last eighteen months fitted with the Ralph Allen 6809 C.P.U. board and running FLEX 09, so I think I am in a fairly good position to make comparisons.

For all the would be performance I would rate the C.P.U.'s in descending order as 6809, Z80, 6502, comparing standard speed versions to be fair (1/2.5/1MHz).

Because the Z80 was designed to retain 8080 compatibility its instruction set is a mess, totally asymetric and needlessly confusing. The Z80 is the only C.P.U. that constantly has me thinking "am I allowed to use this instruction with such and such a register?". The 6502 has a limited but straightforward instruction set, while the 6809 instruction set is far and away the best of any eight bit processor, logical, comprehensive and symetric, and for the record Motorola mnemonics are a model of clarity, putting Zilog and especially Intel firmly in there place (Does anyone actually use the assembler supplied with CP/M? if so, I think you'll know that 8080 mnemonics were obviously the responsibility of the Intel design teams tea boy.)

As regards speed, the Z80 has a faster clock but actually turns out to be slower as it takes an awful lot of cycles to actually do anything. For example Interak 1 using a Z80A at 4MHz takes 1.75 micro-secs to load regester A with immediate data (7 T states at 0.25 micro-secs), while the comparable 2MHz 68B09 takes 1 micro-sec to perform the same operation (2 MPU cycles at 0.5 micro-secs), as, by the way, does the 6502 - so now you know why the Beeb runs faster than the Spectrum! To be fair, the Z80 is the fastest of the three when no memory is accessed, e.g. register to register operations, but just take a look at any typical program and see how much of its time a CPU spends manipulating memory.

For you to say boldly that the 6502 is less powerful than the Z80 is not true, it has less registers and a more limited instruction set and perhaps thats what you meant, but it is straightforward to program and faster (at equivalent clock speeds) at manipulating memory, so your statement should have been qualified by the application you had in mind. At the risk of falling into this trap myself, I'd put the 6809 ahead of both of them for the reasons I've stated.

Regarding bus structures, there are two crucial ommissions from ISBUS. The first is that neither of the interrupt lines are available on the bus, and the lack of IRQ means that the unique and genuinely powerful Z80 feature interrupt mode 2 is not available to Interak users, who are tied

to a particular processor and then denied one of its most powerful features!.

The second is the lack of any provision for memory management or paging. The Tangerine bus has a simple but effective paging circuit built into the motherboard, which means that any one slot of eight may be selected for read and/or occupied by the FDC/RTC and 80x25 Video cards!

The largest card available is the Ralph Adler 128K card, so 6 x 128 = 768k, unfortunately I can't afford all these cards, but the theory still holds! With Interak, a memory card eg DRM-64 is permanently enabled, ie. two cards doesn't equal 128k, just a bus conflict! To impliment CP/M plus I can see only two alternatives, to have some of the 'spare' B side of the edge connector - this would mean track cutting and wire links to suit on existing memory cards, which wouldn't exactly improve reliability apart from being inelegent - or to have a completely new memory card with the MMU on board. The problem here is that the Z80 cannot directly refresh 256k chips, which would be the obvious choice, as they are all to my knowledge 256 milli-secs 8 bit types so they are out unless some sort of refresh control circuitry is added. Perhaps 128k of 4164's could be fitted on to the card along with an MMU, at least then we could use the ports from our DRM-64's.

Anyway, David from Greenbank is a perfectionist in a world full of cowboys and his cards are little short of works of art, which is why I'm still an Interaker. When the 80 column and new CPU cards come out, I'll be confident as always that they work in theory and don't just scrape by in practice providing you use only chips from manufacturer X, but please David, make the new CPU card recognise the interrupt aknowledge cycle as legal (the MZB-3 dosen't), bring IRQ out to the edge connector and give some thought to some sort of paging system.

Yours faithfully
Ian Vaudrey

[Ed- What a super letter. Thank you for taking the trouble to air your views which I will now try to respond to :-
Looking at my diagram of the ISBUS Allocations I see address lines AB16 through AB23 giving an extended address range up to 16 mega-bytes of designed-in expansion. This is supported by the extra control lines of, I/O disable, Memory disable and extended memory request. Whilst the current design of the DRAM-64 does not alow extra cards to be fitted you can be certain that the next step up the memory ladder will do. Refresh is not a problem as by disabling the output buffers during the refresh cycle you can refresh multiple banks of 64k at the same time.
You can be certain that the interupt lines that did not exist on the Kemitron designed MZB-3 card will be included on the first Greenbank design and motherboard lines B4 through B12 have been set aside for this purpose. Priority selection circuitry for interupt handling is built in to the bus on A30 and A31 so all is ready and awaiting a new CPU card. A couple of users have added mode 2 interrupts to the MZB-3 which David describes in IUGN 1 page 7. Take a look at how simple it is to do.
A card using the Z80 has not been designed quite deliberatly. At the moment there is much discusion about the 64180 chip, and I suspect the new card will impliment this chip or one like it. At the same time discarding the Z80 like you would an old but loved motor car. (Total compatability of course must be retained. Would you really swop the clutch with the accelorator when you bought the new car.)
Now to the debate on the relative merits of the various chips.
Assesing the 'power' of a particular chip is not simply a matter of 'how fast any one instruction is'. To me it must be the overall ability of that chip to do useful work for the end user. Consider this argument :-
A computer is a complete entity, from PSU to application software. The Z80 based interak can run CP/M. CP/M gives me the power to use Wordstar, dBase2 etc. Can your 6502 run CP/M?. No it can't. I think you buy a Z80 add on for these computers don't you?
Also the old 8080 ran CP/M, the Z80 runs CP/M and the hitachi 64180 will run CP/M. That means as I develop, all of my software goes with me. I will gladly call the instruction set unstructured but I will never give up my Z80 work-horse and its massive software base for a 6502 or even a 6809. O.K I am being unfair, the 6502 could have been used for CP/M. In which case I would be using that chip. But that is not the real world and I will continue to describe chips that are not supported by there designers in future developments as not as powerful as the 8080/Z80/64180 series, which has run unchallenged for 10 years with lots of time to go. Tell me, where will the 6502 be in 10 years time.
As to numonics, whats wrong with LD A,(addr)? it's very clear to me using Mostec numonics what every instruction does. Never have I wondered "What does this instruction do". Please tell me which numonic causes a problem of not being immediately self-evident?. I totally agree with you about TDL numonics though, and so I either use Macro-80 or ZASM, both of which use Mostec numonics. Your argument about bad numonics must now fail, because whilst a bad set of numonics does exist, another set (MOSTEC) exist, which are probally the most readable that I have ever seen on any computer. Consider MVI A,7 and LD A,7. To me LD A,7 is like reading English, automatic understanding on sight. Just discard TDL numonics and that objection to the Z80 disapears.
Before I go, can I forstall your reaction to my love of CP/M by explaining why I think it is loved by users but hated by salesmen.
CP/M is disliked not because it is no good or user unfriendly but because it is user controllable. I, the user, can manipulate my operating system without having to run to the designer/salesmen each time. Because of this the market rapidly tried to destroy CP/M and get everbody on to much "friendlier" (but expensive) "proper" systems. If you can find me an operating system that I can alter and tailor, one which gives me the power of so much, almost free, software that can run on a 6502 or a 6809 then I might grant you the argument but, until you can the the Z80 wins every time.
Thank you for a most stimulating letter, I hope you will write again to further your case. I propose that you keep this letter for five years or so, and then let us compare the Z80 series to the 6502 series again. oh! I was forgetting they wont be around then, will they!!. But the Interak will be, and I'll bet it will be running a Z80 compatible MPU.]

==========

CHARLIE'S LIB.

Many thanks to Bruce Joyce, Mel Saunders and F.R.Johnson for their contributions to the Disk Libary. These were very much appreciated. This issue adds another four disks to the Libary, IUG 4-7, and barring any unforeseen disasters there should be another three disks issued with the next newsletter. To date eighteen disks have been sent out and no complaints have been received. Don't forget this is your Libary, so if you have any programs to contribute please let us have them together with any suggestions, requests or complaints.

SCAN.COM on IUG-5 can be highly recommended. Having a poor memory I have found it invaluable. Before obtaining it I used to spend hours going through back issues of magazines looking for an article or circuit that I knew I had seen in the past. Its use is not confined to single line references as provided the same keyword is put on each line it will return all the lines required in order. You can also use it to scan a dictionary, or the location of components or other items. In fact its use is limited only by the imagination.

Having spent an enjoyable day at the Radio Society of Great Britain Rally at Drayton Manor Park, Staffs, recently I would like to draw the attention of all hardware enthusiasts to these RSGB rallies. The trade stands cover everything from amateur radio,CCTV, electronic components and computers and many surplus dealers attend them. There are many bargains to be picked up though, of course, you have to be a little wary of the second-hand goods. I have a spare Shugart 8 inch disk drive with a 115V motor so when I saw a Shugart 800 with a 240V motor for 5 pounds I could'nt resist, thinking that I would use the motor at least. However, before changing over the motors I cleaned it up and tried it with the Interak Format program. It sailed through to track 76 with flying colours so I now have two spare eight inch drives although one still has a 115V motor. These rallies are well worth a visit although they do vary a lot both in size and content. Some are held in Civic Centres, some in small halls and some like the Drayton Manor Park one are very large and the variety of traders who attend varies with the size. In case anyone is interested, a list of future rallies is shown below. As I am not sure when this newsletter will be circulated the list runs from June to the end of the year. They are open to the public and you do not have to be a RSGB member to attend.

14 June Elvaston Castle Country Park. Nr Derby.
14 June RNARS Rally HMS Mercury Nr Petersfield. Hants.
21 June Denby Dale Mobile Rally. Shelley High School, Nr Huddersfield.
28 June Longleat Rally. Longleat Park, Nr Warminster.
12 July Worcester & DARC Droitwich Rally High School Droitwich.
19 July Cornish Mobile Rally. Camborne College of Further Education.
19 July McMichael 87 Rally. Haymill Youth & Community Centre, 112 Burnham Lane, Slough.
26 July Scarborough ARS Rally. The Spa, Scarborough.
2 August RSGB Mobile Rally. Woburn Abbey, Woburn, Bedfordshire.
2 August Rolls-Royce ARC Rally. Rolls-Royce Sports & Social Club Barnoldwick.
9 August 30th Derby Mobile Rally. Lower Bemrose School, St Albans Road, Derby.
9 August Hamfest 87 & Craft Fair Wimborne, Dorset.
16 August Red Rose Rally. Bolton Sports & Exhibition Centre, Bolton.
6 September Preston ARS 20th Annual Rally. Lancaster University.
13 September Lincoln Hamfest, Lincolnshire Showground, Lincoln.
13 September Scottish AR Convention. The Magnum Sports & Leisure Centre, Irvine, Ayrshire.
13 September National Amateur.Radio Car Boot Sale, Old Warden Aerodrome, Beds.
13 September SMC Open Day, Chandler's Ford Industrial Estate, Eastleigh, Hants.
13 September Telford Mobile Rally. Telford Racquet & Fitness Centre, Telford.
20 September Peterborough R & ES Rally . Wirrina Sports Stadium, Peterborough.
20 September Trafford Rally & Components Fair, Lancs CCC (Old Trafford), Talbot Road, Stretford, Manchester.
20 September Vange ARS Rally, Nicholas School, Leinster Road, Laindon.
27 September Harlow Mobile Rally, Harlow Sports Centre.
4 October Wakefield Mobile Rally.
4 October Great Lumley AR & ES Rally, The Community Centre, Great Lumley, Chester-le-Street, County Durham.
22 November West Manchester RC Winter Rally, Pembroke Halls, Walkden.

That's all for now. Happy rallying!

Charlie.

DISK LIBRARY INDEX - UPDATE PAGES

Please amend your disk library index as follows :-

Replace page IUG-1 with new page IUG-1 enclosed.
Replace page IUG-2 with new page IUG-2 enclosed.
Replace page IUG-3 with new page IUG-3 enclosed.
Insert new page IUG-4 enclosed.
Insert new page IUG-5 enclosed.
Insert new page IUG-6 enclosed.
Insert new page IUG-7 enclosed.

IUG 1 - 588k

F83	.COM	24K	FORTH-83 FOR CP/M BY PERRY & LAXEN.
README	.80	20K	F83 INSTRUCTIONS. DOC
F83-FIXS	.TXT	8K	F83 VERSION 1.0 UPDATE.
BASIC	.BLK	20K	BASIC COMPILER IN F83.
CLOCK	.BLK	12K	SOURCE FOR A CALENDAR EXAMPLE.
CPU8080	.BLK	44K	8080 DEPENDENT CODE.
EXPAND80	.BLK	8K	ORIGINAL SOURCE TO EXPAND.HUF.
EXTEND80	.BLK	32K	EXTENSIONS SOURCE.
HUFFMAN	.BLK	44K	COMPRESSION PROGRAM.
KERNEL80	.BLK	108K	KERNEL SOURCE.
META80	.BLK	52K	METACOMPILER SOURCE.
UTILITY	.BLK	112K	UTILITY SOURCE.
USQ	.COM	4K	UNSQEEZES SQUEEZED FILES.
LISP	.COM	20K	UPDATED LISP.
INITLISP	----	4K	
INITLISP	.STB	4K	
LISP	.DOC	16K	INSTRUCTIONS.

NOTES

==========

****
* README.80 PRINTED IN IUGN-16 *
* LISP.DOC PRINTED IN IUGN-16 *
****

==========

FORTH LANGUAGE RELATED FILES

F83.COM	24K	FORTH-83 BY PERRY & LAXEN.
META80.BLK	52K	
KERNEL80.BLK	108K	
EXTEND80.BLK	32K	
CPU8080.BLK	44K	
UTILITY.BLK	112K	
HUFFMAN.BLK	44K	
CLOCK.BLK	12K	
F83-FIXS.TXT	8K	
README.80	20K	
EXPAND80.BLK	8K	

Read the file README.80 for information on the start up and use of FORTH.

The following books are helpful in using FORTH-83.
Inside F83 by C.H.Ting.

Mastering Forth by Anderson & Tracy.

Forth. A text and reference by Kelly & Spies.

WELCOME TO FORTH

F83<CR> (fire up forth from the CP/M environment)
8080 Forth 83 Model 2.0.0 (Forth's reply and)
Modified 01Apr84 (sign on message)

Standard System Documentation Requirement

1. The system dictionary space is CPU dependant and can be determined by typeing HERE U. <CR>

2. The application dictionary space is also CPU dependant and is the difference between the top of the dictionary and the beginning of the parameter stack. The location of the parameter stack varies depending on the amount of memory available to the machine. The application dictionary space can be determined by typing SP0 HERE -U. <CR>

3. The data stack space is the same as the dictionary space.

4. The return stack space was arbitrarily set at 256 bytes. It can be altered by remeta-compiling the system.

5. No mass storage block ranges are reserved by the system, other than the contents of the files that are distributed.

6. The user has available to him blocks 0 thru 65534. Note that block 0 may not be used for loading. Block number 65535 is reserved to indicate the buffer is missing.

7. Any ascii terminal should work with this system. If the user has a cursor addressable terminal, the editor can be easily modified to take advantage of the terminal's characteristics.

==========

LISP LANGUAGE RELATED FILES

LISP	.COM	20K	UPDATED LISP.
INITLISP	----	4K	
INITLISP	.STB	4K	
LISP	.DOC	16K	INSTRUCTIONS.

Read LISP.DOC for info on running the Interpreter.

Artificial intelligence language invented by mathematicians and engineers. In LISP you start with a master sentence such as :-

PLAY CHESS(BOARD,PIECES,MOVES)

and then further define the statements in brackets until a complete program exists.

The following books may be of value :-

"Let's talk LISP" by Laurent SIKLOSSY
Prentice Hall, Inc., 1976
"LISP" by Patrick H.WINSTON & Berthold K.P.HORN
Addison-Wensley Pub.Co., 1981
"Anatomy of LISP" by John ALLEN
McGraw Hill Book Co., 1978.

HOW TO RUN LISP
Capital letters must be used for all LISP reserved words (essentially names of functions); because of this it is recommended that capital letters be used throughout.

To run the interpreter proceed as follows :
1. Type LISP followed by a <cr>.
2. Wait for the message R E A D Y to appear ; this indicates that the interpreter is ready to evaluate a LISP expression.
3. Type the LISP expression to be evaluated on one or more successive lines (additional blanks do not produce any effect).
   The expression must be preceeded by a left parenthesis '(' and followed by a right parenthesis ')'.
   After having entered the LISP expression, type twice <cr> to initiate the evaluation.
   (Ensure that the correct number of parentheses is entered otherwise the interpreter will refuse to start the evaluation as it expects more input data).
4. Once the evaluation is completed, the interpreter will print out the resulting expression.
5. After that, other evaluation cycles can be introduced by repeating sequentially steps 3 and 4.

To terminate the process and return to CP/M type FIN followed by two <cr>'s.

INITLISP is the name of a file which must be available on the same diskette as the interpreter for the latter to work properly.
Upon initiation of the interpreter, the contents of INITLISP is read and used to alter the LISP environment by adding user's defined variables (with their values) and/or functions.
INITLISP actually contains elements of a program written in LISP and any of the characters employed to write a LISP program may be used. However tabs (and perhaps other keys not representing alphanumeric characters, dots, commas, spaces, etc) should be avoided as they may produce unexpected results. Further, INITLISP must be terminated by the atom FIN

followed by two <cr>'s to avoid reading beyond the eof.
INITLISP as presented only contains the atom FIN followed by two <cr>'s. In other words INITLISP will not affect at all the LISP environment upon activation of the interpreter.
The file INITLISP.STB, also included for the purpose of demonstration, contains a number of functions directly written in LISP. After renaming this file INITLISP (after having saved the original INITLISP for possible re-use), all these functions will become integral part of the LISP environment upon initiation of the interpreter. Hence they can be utilized as they were actual intrinsic function (with some loss of efficiency).

The following terms apply :

a. SEX
b. ATOM
c. DOTTED-PAIR
d. LIST
e. SPECIAL ATOMS T=TRUE, NIL=FALSE

Atoms are always taken as representing variables unless they are preceeded by the atom QUOTE. In the first case the interpreter attempts to evaluate them (in other words looks for the value associated with them) while in the second case it does not do that.

The list of the functions which are supported by the interpreter follows :-

AND,APPEND,ATOM,CAR,CDR,CONS,COPY,
DEFEXP (DEFine EXPression),
DEFFEXP (DEFine F-EXPression),EQ,EQUAL,EVAL,LAST,
LENGTH,LIST,NOT,NULL,OR,PROG2,PROGN,QUOTE,
REMOB (REMove OBject),REPLACEH (REPLACE Head),
REPLACET (REPLACE Tail),REVERSE,SET,
SETQ (SET Quote),TRACE,UNTRACE

The following additional features are included :

PROG control structure, allowing iterative programming style (limitations : GO and RETURN cannot be used to jump outside PROG);
CONDitional expressions;
LABEL for temporary function definitions;
LAMBDA, FLAMBDA to introduce un-named functions evaluating (respectively non-evaluating) their arguments;
FUNCTION construct to deal with with the FUNARG problem.

==========

IUG 2 - 548k

CINTERP	.COM	16K	COBOL INTERPRETER.
COBOL	.COM	16K	N.P.S. MICRO COBOL VER 2.1
EXEC	.COM	8K	" " " "
PART 2	.COM	16K	" " " "
COBOL	.DOC	40K	INSTRUCTIONS.
ADD	.CBL	4K	COBOL PROGS. SEE COBOL.DOC
CBL1	.CBL	4K	" " " " "
CBL2	.CBL	4K	" " " " "
DEMO	.CBL	4K	" " " " "
SEQ	.CBL	4K	" " " " "
DEMO	.CIN	4K	" " " " "
DEMO	.LST	4K	" " " " "
CBL1	.CIN	4K	" " " " "
CBL1	.LST	4K	" " " " "
CBL2	.CIN	4K	" " " " "
CBL2	.LST	4K	" " " " "
CBL1	.FIL	4K	" " " " "
ADD	.CIN	4K	" " " " "
ADD	.LST	4K	" " " " "
ZSC-1	.C	20K	SMALL-C COMPILER.
ZSC-2	.C	20K	SMALL-C COMPILER.
ZSC-COMP	.LIB	12K	" "
ZSMALL	.COM	24K	" "
ZSMALL	.DOC	16K	INSTRUCTIONS.
C-UTIL	.DOC	4K	DOC. ON TAB, LIST, FILECHOP, AND UNLOAD.
CONIO	.LIB	8K	SMALL-C LIBARIES.
CRUN	.LIB	8K	" "
CZMON	.C	12K	SAMPLE SMALL-C PROGRAM.
CZMON	.COM	8K	" " "
FILE	.LIB	12K	SMALL-C LIBARY.
FILECHOP	.C	4K	CHOP LARGE FILES INTO SECTIONS.
FILECHOP	.COM	4K	
LIST	.C	4K	SAMPLE SMALL-C PROGRAM.
LIST	.COM	4K	
NUMIO	.LIB	4K	SMALL-C LIBARY.
TAB	.C	4K	SAMPLE SMALL-C PROGRAM.
TAB	.COM	4K	
Z80ASMUK	.COM	12K	Z80 ASSEMBLER FOR USE WITH SMALL-C.
Z80DOCUK	.DOC	8K	INSTRUCTIONS.
ADV	.COM	36K	EXPANDED ADVENTURE GAME.
ADVT	.DOC	4K	INSTRUCTIONS.
ADVI	.DAT	32K	
ADVI	.PTR	4K	
ADVT	.DAT	100K	
ADVT	.PTR	16K	

NOTES

==========

CINTERP	.COM	16K	COBOL INTERPRETER.
COBOL	.COM	16K	N.P.S. MICRO COBOL VER 2.1
EXEC	.COM	8K	" " " "
PART 2	.COM	16K	" " " "
COBOL	.DOC	40K	INSTRUCTIONS.
ADD	.CBL	4K	COBOL PROGS. SEE COBOL.DOC
CBL1	.CBL	4K	" " " " "
CBL2	.CBL	4K	" " " " "
DEMO	.CBL	4K	" " " " "
SEQ	.CBL	4K	" " " " "
DEMO	.CIN	4K	" " " " "
DEMO	.LST	4K	" " " " "
CBL1	.CIN	4K	" " " " "
CBL1	.LST	4K	" " " " "
CBL2	.CIN	4K	" " " " "
CBL2	.LST	4K	" " " " "
CBL1	.FIL	4K	" " " " "
ADD	.CIN	4K	" " " " "
ADD	.LST	4K	" " " " "

COBOL is a programming language for solving business problems such as calculating payrolls, or maintaining sales ledgers. COBOL stands for COmmon Business Oriented Language.
A book with a large section on using the COBOL language is :-
Computer Programming Made Simple by J.Maynard.
Publisher W.H.Allen. See section 10.

NPS MICRO-COBOL Ver. 2.1
Read COBOL.DOC for fuller information.

This compiler is composed of two parts, each of which reads a portion of the input file. Part One reads the input program to the end of the Data Division and builds the symbol table. At the end of the Data Division, Part One is overlayed by Part Two which uses the symbol table to produce the code. The output code is written as it is produced to minimize the use of internal storage.The EXEC Program builds the core image for the intermediate code and performs such functions as backstuffing addresses and offsetting address in subroutines. EXEC then copies the interpreter (CIN-TERP.COM) into memory and transfers control to it. The interpreter is controlled by a large case state-ment that decodes the instructions and performs the required actions.

==========

THE SMALL C LANGUAGE

ZSC-1	.C	20K	SMALL-C COMPILER.
ZSC-2	.C	20K	SMALL-C COMPILER.
ZSC-COMP	.LIB	12K	" "
ZSMALL	.COM	24K	" "
ZSMALL	.DOC	16K	INSTRUCTIONS.
C-UTIL	.DOC	4K	DOC. ON TAB, LIST, FILECHOP, AND UNLOAD.
CONIO	.LIB	8K	SMALL-C LIBARIES.
CRUN	.LIB	8K	" "
FILE	.LIB	12K	SMALL-C LIBARY.
NUMIO	.LIB	4K	SMALL-C LIBARY.

Read ZSMALL.DOC for instructions and info.

Small C is a subset of the standard C language. It was written by Ron Cain and originally published in Dr.Dobbs Journal of Computer Calisthenics and Orthodontia, No.45. The version described has been modified to generate Z80 mnemonics instead of the original 8080 ones. A number of bug fixes, most of them sent in by readers of DDJ, have also been incorporated.
The compiler inputs a program written in Small C from a file and produces a version of the program in Z80 assembler language mnemonics which can be assembled and run on a Z80 system. The original run time library has been supplemented by some additional routines which use CP/M I/O facilities. The library is divided into modules so that routines that are not needed for particular applications can be omitted.
The original Small C program published by Ron Cain was itself written in C.
The version described here was typed into an Onyx in the Electrical Engineering Department at Brighton Polytechnic and compiled on the Onyx's C compiler to produce a Small C compiler running under UNIX. A slightly modified version ofthe Small C source was then fed into the Onyx's Small C compiler to produce a Z80 assembler file which was assembled. The final Z80/8080 code was then moved to a microcomputer and hence to a floppy disc. This version has been further modified under CP/M. As far as is known the slight eccentricities of Z80ASM have been accomodated. The output is also compatible with other normal Z80 assemblers.

References.

Dr. Dobbs Journal of Computer Calisthenics and Orthodontia, Box E, Menlo Park,CA 94025

No. 45 (May 1980) "A Small C Compiler for the 8080's" by Ron Cain. A listing in C of the 8080 compiler + notes on implementation. This issue also contains other articles about C.

No. 48 (Sept. 1980) "A Runtime Library for the Small C Compiler" by Ron Cain. Run time library + useful notes. (See also Nos. 52, 57 and 62 for bug notes by P.L.Woods, K.Bailey, M.Gore & B.Roehl, J.E.Hendrix)

Nos.81 82 (July, Aug. 1983) "RED - A better Screen Editor" by E.K.Ream. A screen editor written in Small C with a number of library routines.

"The C Programming Language" by B.W.Kernighan & D.M.Ritchie (Prentice-Hall).
The standard text on C.

==========

FILECHOP.C SAMPLE SMALL-C PROGRAM.
FILECHOP.COM
This is a utility written in Small C.
FILECHOP Chops large files into sections.
The program prompts for the input file name, the size of the file sections, and asks if the file is to be treated as a text file. If the file is not a text file, the break is at the exact number of kbytes requested. Text files are cut after the next CRLF and have ^Z added at the end of each section. Output file names are numbered 00,01,etc with the same type as the original input file. The names will be truncated if they are more than 6 characters long to give room for the numbers. File names must be given in full with drive names if needed. Output files are sent to the same disk as the input file and it is up to the user to see that there is room for them.
The command line for FILECHOP.COM is simply:-
[Drive:]FILECHOP

CZMON.C 12K SAMPLE SMALL-C PROGRAM.
CZMON.COM 8K
A Z80 monitor program allowing Set breakpoint, Copy memory, Display disk dir, Execute program, Input/Output to a port, Read and Write files plus many other facilities.

LIST.C 4K SAMPLE SMALL-C PROGRAM.
LIST.COM 4K
This is a utility written in Small C.
LIST lists an ASCII file to the console 20 lines at a time and then pauses for a keypress. The 20 line block is modified if there are any very long lines that extend to several lines on the screen.
The command line for executing LIST.COM is:
[Drive:]LIST filename

TAB.C 4K SAMPLE SMALL-C PROGRAM.
TAB.COM 4K
This is a utility written in Small C.
TAB converts an Intel format file such as the HEX files produced by many assemblers into a tabulated hex listing. The output can be directed to CON:, LST: or another file. There is a built-in assumption that the lines of input are limited to a maximum of 16 bytes of encoded data. The input and output file names must be given in full and may include a drive. The filenames are separated by '>'.
The command line for executing TAB.COM is :-
[Drive:]TAB [Drive:]Input_file>[Drive:]Output_file

Z80ASMUK.COM
Z80DOCUK.DOC
Read Z80DOCUK.DOC for info

Z80ASM - ASSEMBLY LANGUAGE PROCESSOR.
Z80ASM is an assembler for the Z80-CPU Microprocessor, using the ZILOG/MOSTEK Mnemonics. It is designed to run under the CP/M 1.4 or 2.2 Operating systems from Digital Research. CP/M will run on either an 8080 or a Z80 system.

Z80ASM reads a source (.ZSM) file created using the CP/M 'ED' or similar Text Editor. The source may be in either Upper or Lower case letters. The program produces an optional Object code (.HEX) file in INTEL HEX format, and a listing sent either to the Console (CON:) device or the (LST: device), or to a print (.PRN) file. The 'HEX' file can be loaded for execution using 'DDT' or 'LOAD' utilities of CP/M.

The PSEUDO-OPERATORS available in Z80ASM are :-

```
<LABEL> EQU   <EXPRESSION>  ASSIGN VALUE TO LABEL
        ORG   <EXPRESSION>  SET PROGRAM COUNTER
        DEFS  <EXPRESSION>  RESERVE STORAGE
        DEFW  <EXPRESSION>  DEFINE WORD (2 BYTES)
        DEFB  <EXPRESSION>  DEFINE BYTE (FORMAT 1)
        DEFB  '-STRING-'    DEFINE BYTE (FORMAT 2)
        EJECT               LISTING EJECT
        END   <EXPRESSION>  DEFINE EXECUTION ADDR
```

NOTE: for DEFW only one Operand is allowed. While for DEFB mixed <EXPRESSION> and 'STRINGS' separated by commas are permitted.

EXPRESSIONS:
Expressions may be any number of the items listed, separated by + - * and / for addition, subtraction, multiplication and division respectively.
The Division supported is an unsigned 16-bit by 8-bit integer divide with the Quotient only being returned, the remainder is discarded.
Brackets within expressions are not permitted, but if used will be flagged as errors.
BOOLEAN Operators are NOT permitted, though if used these probably will not be flagged as an error, and the statement will not be assembled correctly.

EXPRESSION ELEMENTS:

```
  <LABEL>       1 TO 11 CHARACTER STATEMENT LABEL
  <NUMBER>H     HEXADECIMAL NUMBER
  <NUMBER>O     OCTAL (BASE 8) NUMBER
  <NUMBER>Q     OCTAL (BASE 8) NUMBER
  <NUMBER>B     BINARY (BIT) NUMBER
  <NUMBER>      DECIMAL NUMBER
    $           PROGRAM COUNTER REFERENCE
   'X'          X = ANY PRINTABLE CHARACTER
   'XX'         AS ABOVE, BUT FOR 16-BIT REGISTER
```

EXAMPLES OF EXPRESSIONS ARE:

```
        LABEL1+5
        0FFH
        420
        200
        10100001B+6
        $-6
        '9'+1
        LD HL,25*10
        LD A,44/11
        LD HL,'AB'
```

The source file must have an extension of 'ZSM'. The object file will have the extension 'HEX' and the print file that of 'PRN' if they are created.

Z80ASM is called using the following format :-

Z80ASM <FILENAME>.<ABC> or Z80ASM n:<FILENAME>

```
ADV         .COM    36K   EXPANDED ADVENTURE GAME.
ADVT        .DOC     4K   INSTRUCTIONS.
ADVI        .DAT    32K
ADVI        .PTR     4K
ADVT        .DAT   108K
ADVT        .PTR    16K
          Read ADVT.DOC for information.
```

This is an expanded version of Adventure. This one has a cave that is twice as large (it's a 550 point version). All of the features of the original adventure are in this version, plus a whole bunch of new rooms, treasures, and ways for the bumbling explorer to get oneself killed.
Adapted from the original version of Adventure by Mike Goetz.

```
*                                          *
*  COBOL.DOC          These four document files  *
*  ZSMALL.DOC         all printed in IUGN-17     *
*  Z80DOCUK.DOC       Issued September 1987.     *
*  ADVT.DOC                                      *
*                                          *
```

IUG 3. - 568k

BCKUP	.COM	4K	DISK BACKUP PROGRAM.
COMPARE	.COM	4K	BINARY FILE COMPARISON.
COPY	.COM	4K	DISK COPY PROGRAM.
CPACK	.DOC	12K	DOCUMENTATION FOR BCKUP, COMPARE, RESTORE, COPY AND SORTDIR.
DOS	.COM	4K	FINDS ADDRESS OF CCP/BDOS & SIZE OF TPA.
RESTORE	.COM	4K	RESTORES ERASED FILES.
SORTDIR	.COM	4K	SORTED DIRECTORY PROGRAM.
DUTIL	.COM	12K	REVISION OF DISK UTILITY WITH EXTENDED FEATURES.
XLATE2	.COM	8K	TRANSLATES INTEL 8080 SOURCE TO ZILOG Z80 CODE.
DIRSCAN	.COM	8K	SCANS DIRECTORY.
INDEXER	.COM	20K	CREATES AN INDEX FOR A BOOK OR ANY DOCUMENT
INDEXER	.SUB	4K	AUTOMATICALLY. INCLUDES A SAMPLE PROGRAM.
PZKEY	.INX	4K	
PZKEY	.TRE	4K	
INDEXER	.DOC	16K	DOCUMENTATION.
OKI	.COM	8K	MENU FILE TO SEND CODE TO OKIDATA 82/83.
EPSON	.COM	8K	MENU FILE TO SEND CODE TO EPSON MX PRINTERS TO SET TYPE SIZE.
XDIR	.COM	4K	EXTENDED DIRECTORY.
SUB	.COM	16K	MENU FILE TO RUN THE MAJOR CP/M COMMAND FILES.
ERASE	.COM	8K	"USER FRIENDLY" ERASE.
SIGNS	.COM	12K	FORMATTING PROGRAM FOR BOTH 80 AND 132 COLUMN PRINTERS.
SIGNS11	.COM	12K	
SIGNS	.TXT	4K	
SIGNS	.DOC	4K	
SIGNS6	.COM	12K	
FONT	.DAT	4K	
DELBR	.COM	16K	TO EXTRACT .LBR FILES TYPE DELBR FILENAME.
DELBR11	.COM	16K	EXTRACTS .LBR FILES CP/M80, 86 AND MSDOS.
FTNOTE13	.COM	16K	PRODUCES FOOTNOTES WITH WORDSTAR.
FTNOTE13	.DOC	24K	INSTRUCTIONS.
QK12	.COM	4K	REDEFINES KEYBOARD.
QK12	.BUG	4K	" "
QK12	.DOC	12K	DOCUMENTATION.
BASFK	.ASM	12K	ROUTINE TO LOAD CIFER VDU FUNCTION KEYS WITH BASIC STATEMENTS.
CAT	.COM	4K	CATALOGUE SYSTEM.
CAT2	.COM	4K	" "
CRCK	.COM	4K	CHECKSUM PROGRAM.
CRCKLIST	.CRC	4K	CHECKSUM OF SOME FILES ON THIS DISK.
DDISK	.COM	8K	IMPROVED DISK DEBUGGER
DDISK	.MAC	36K	SOURCE OF THE ABOVE
MAST	.CAT	4K	SAMPLE CATALOGUE FILE.
PRTHT/21	.COM	4K	PRINT LISTINGS WITH DATE AND TIME.
PRTHT/21	.ASM	32K	SOURCE OF ABOVE.
PRTHT	.DOC	4K	DOCUMENTATION.
PWS/5	.ASM	32K	WORDSTAR PATCHER FOR INTELLIGENT TERMINALS/PRINTERS
PWS	.DOC	4K	INSTRUCTIONS FOR ABOVE.
TESTZ80	.ZSM	4K	TEST SOURCE FILE FOR Z80 ASSEMBLER.
Z80ASMUK	.ASM	60K	IMPROVED Z80 ASSEMBLER.
Z80ASMUK	.COM	12K	" " "
Z80DOCUK	.DOC	8K	DOC FOR ASSEMBLER.
UDCAT	.COM	12K	IMPROVED DISK CATALOGUE PROGRAM.
UDCAT	.MAC	20K	
UDCAT	.DOC	8K	DOCUMENTATION FOR ABOVE.

NOTES

CPACK.DOC DOCUMENTATION FOR BCKUP, COMPARE, RESTORE, COPY AND SORTDIR.

CPACK is a package of utility programs designed for use with the CP/M operating system. It contains a number of programs designed to make use of the system easier, in particular for those systems using a single disk drive which are not well catered for by the CP/M software as it is distributed.

The following programs are described in this document file :-

BACKUP - Disk copier
COMPARE - File comparison utility
COPY - File copier
RESTORE - File recovery program
SORTDIR - Directory sorter

You may find that the file BACKUP mentioned in this document has been renamed BCKUP on this disk in order to avoid confusion with another program with a similar name.

BCKUP.COM DISK BACKUP PROGRAM.
CPACK.DOC INSTRUCTIONS FOR USING BCKUP.COM

BACKUP is a program for making backup copies of an entire diskette on a single drive system. BACKUP prompts the user to insert source and destination disks. The disk contents are copied on a track to track basis, buffered in memory.

COMPARE.COM BINARY FILE COMPARISON.
CPACK.DOC INSTRUCTIONS FOR USING COMPARE.COM

A>COMPARE <sourcefile> <compfile>
Carries out a byte for byte comparison between two files. If any difference is found, the following information is presented on the console:
The record number in the file
The byte number within the record
The ASCII and hex values of the differing bytes

COPY.COM DISK COPY PROGRAM.
CPACK.DOC INSTRUCTIONS FOR USING COPY.COM

A>COPY <sourcefile> <destfile>
A>COPY <sourcefile>
A>COPY

Copy is a program to simplify disk to disk copies of files in CP/M systems having a single drive. It operates in a somewhat similar manner to the CP/M supplied utility PIP, the disk file copying routines of which are not usable in single drive systems.
It operates by buffering the file into memory and then prompting for a disk change before dumping the file onto the new disk.

RESTORE.COM RESTORES ERASED FILES.
CPACK.DOC INSTRUCTIONS FOR USING RESTORE.COM

A>RESTORE <filename>
A>RESTORE <filename> n

RESTORE is a program to recover CP/M files which have been inadvertently ERAsed. CP/M erases files by setting a byte in their disk directory entry, the actual file data is not erased but merely becomes eligible for being overwritten. RESTORE resets the byte, making the file once again useable.
WARNING: Restore can have some strange and unpredictable effects if it is not used properly. It should only be used after a file has been erased and before any other operations affecting the disk directory (such as SAVE, Editing, COPY, etc) have been carried out. The reason for this is that these operations may have utilised part of the disk space of the ERAsed file.

SORTDIR.COM SORTED DIRECTORY PROGRAM.
CPACK.DOC INSTRUCTIONS FOR USING SORTDIR.COM

A>SORTDIR
A>SORTDIR *

SORTDIR is a program which will sort the disk directory into alphabetical order, either by filename or by file type (ASM, COM, etc) and name. The second command line above sorts by file type.
In addition to sorting the directory, SORTDIR also tidies the directory entries by completely deleting all ERAsed files and all zero length files. These files have their complete directory entries overwritten. Use of SORTDIR is recommended after RESTORE has been used and all files have been checked.

DUTIL.COM

REVISION OF DISK UTILITY WITH EXTENDED FEATURES.
A very powerful disk debbugger that lets you view and alter any on disk data found by sector and track number. You can recover erased files by altering the disk directory, dump damaged areas and rewrite them back to disk, examine program code "on-disk", you can even inspect and alter the operating system tracks, perhaps to alter the loader code. Dutil can provide a disk map to indicate which sector and tracks are used by a particular file, this can let you move the good sectors from an unreadable file so that you only have to recreate the one duff sector again. It is a good idea, at first usage, to switch on the printer with CTRL P before entering Dutil, this will produce a hardcopy of the help menus for later reference.

INDEXER.COM
INDEXER.SUB
INDEXER.DOC
PZKEY.INX
PZKEY.TRE

CREATES AN INDEX FOR A BOOK OR A DOCUMENT AUTOMATICALLY. IT INCLUDES A SAMPLE PROGRAM.

An index is a table that lists the important topics of a book in alphabetical order, showing for each the numbers of the pages on which that topic is mentioned. An index is an indispensable part of any non-fiction book. Even a poor index can give better access to a book than a table of contents can, while a good index increases the utility of a book many times.
The INDEXER program is a machine aid for a human indexer. It does the job of the pile of index cards, and it makes the finished index automatically, as a disk file that can be edited or printed.

XDIR.COM

A>XDIR or A>XDIR B: etc
Produces a better directory listing than DIR.
Arranged in A-Z order with file sizes.

SIGNS.COM
SIGNS11.COM
SIGNS.TXT
SIGNS.DOC
SIGNS6.COM
FONT.DAT

SIGNS prints a string of letters in large format - 10 x 7 normal characters. Each letter is 12 x 7 characters in size. There are two spaces between each letter, and two line feeds between each string. Two compilations are provided; SIGNS6 has six letters per string and is suitable for consoles and 80 column printers. SIGNS11 has 11 characters per line which fits onto a 132 column printer.

FTNOTE13.COM PRODUCES FOOTNOTES WITH WORDSTAR.
FTNOTE13.DOC INSTRUCTIONS.

FTNOTE.COM is a formatting utility for WordStar files. Its main function is to allow the printing of WordStar documents with page-bottom footnotes, but it does internal page referencing and improved block formatting as well. It will automatically move your footnotes into and out of your text file; number them for use as end notes; or produce a printable file with them at page bottom. It is an effective but simple program. It does NOT attempt to duplicate the existing capabilities of WordStar: editing, justification, or printing. It is a utility to be used in conjunction with WordStar.

QK12.COM REDEFINES KEYBOARD.
QK12.BUG
QK12.DOC DOCUMENTATION.

QwikKey allows console keys to produce specified character strings. Keys may be defined at any time, even while you are running a program like DDT, MBASIC, WordStar, and most others. Key definitions remain across warm boots, but are lost on cold boots. The utility is meant to be used on-the-fly, to define commands as you need them, rather than predefining a set of keys for a particular program.
QwikKey allows the user to assign character strings to keys. When a key having a string defined in this way is struck, the defined string, rather than the character normally associated with the key, is delivered to the program running at the time. Key definitions may be loaded from files containing previously saved definitions, or they may be defined on-the-fly, even while a program is running. Both normal keys (i.e. keys generating a single character) and keys generating escape sequences are supported. The maximum length of the defined string is 31 minus the length of the character or string normally generated by the key in question. Thirty-one different keys may be defined.

UDCAT.COM IMPROVED DISK CATALOGUE
UDCAT.MAC
UDCAT.DOC DOCUMENTATION FOR ABOVE.
CAT.COM
CAT2.COM
MAST.CAT

CP/M DISK CATALOGUING SYSTEM
UDCAT.COM Maintains the master catalogue MAST.CAT
CAT.COM Allows a 'R' type search of JST.CAT'.
CAT2.COM pes' the entire sorted JST.CAT'
MAST.CAT An embryo master catalogue

Maintains a disk catalogue of all files on all disks. It can be scanned and maintained and if Charlie would produce the missing file CATP.COM, the catalogue could be printed out as a record of what is where.

IUG-4 560K

ATTEN	.COM	16K	ATTENUATOR DESIGNER.
AUTOLOAD	.COM	12K	AUTO START FUNCTION AFTER
AUTOLOAD	.DOC	4K	... COLD BOOT.
BAUDSET	.ASM	8K	UTILITY TO ALTER BAUD RATES OF
BAUDSET	.DOC	4K	... I/O DEVICES. H/W DEPENDENT.
CHESS643	.COM	12K	CHESS WITH FULL GRAPHICS
CHESS643	.DOC	4K	... NEEDS A VDU2K.
COUNT	.COM	4K	COUNTS WORDS AND LINES IN AN
COUNT	.DOC	4K	... ASCII FILE. OK WITH WS.
CPMFTH	.ASC	4K	FORTH UTILITY TO MAKE FORTH
FTHCPM	.DOC	8K	... FILES FROM CPM FILES.
FTHCPM	.SCR	20K	-- " --
HEADER6	.COM	24K	
INVADA	.COM	8K	SPACE INVADERS FOR VDU2K.
JHELP	.COM	4K	
MX80NG	.COM	4K	
PBH	.ASM	20K	PRINTS BLOCK TITLES
PBH	.COM	4K	-- " --
POWER	.COM	12K	
RBBS-USE	.DOC	12K	BULLETIN BOARD PACKAGE
RBBS31	.ASC	20K	-- " --
RBBS31A	.DOC	12K	-- " --
RBSUTL31	.COM	24K	-- " --
SCRAMBLE	.COM	4K	FILE PROTECTION UTILITY.
SCRAMBLE	.DOC	4K	-- " --
SETC8SD	.ASM	4K	ALLOW DRIVE C TO BE 8" WITH
SETC8SD	.COM	4K	... A & B AS 3.5 OR 5.25".
SUPERSUB	.ASM	24K	ADVANCED SUBMIT.COM
SUPERSUB	.COM	4K	-- " --
SUPERSUB	.DOC	16K	-- " --
SYNONYM3	.ASM	12K	CP/M UTILITY.
SYNONYM3	.DOC	8K	-- " --
TAGCPM	.BAS	20K	MODEL TRAIN TRAFFIC CONTROL
TRAFFIC	.TXT	4K	-- " --
TEMP	.COM	8K	TEMPERATURE CONVERSION
TTHELP0	.DAT	4K	TOUCH TYPEING TUTOR
TTHELP1	.DAT	4K	-- " --
TTHELP2	.DAT	4K	-- " --
TTHELP3	.DAT	4K	-- " --
TTHELP4	.DAT	4K	-- " --
TTHELP5	.DAT	4K	-- " --
TTKEYBD	.DAT	4K	-- " --
TTYPE	.BAS	24K	-- " --
TTYPE	.DOC	20K	-- " --
TTYPE	.OBJ	24K	-- " --
TTYPE	.WST	20K	-- " --
TTYPEXA	.DAT	4K	-- " --
TTYPEXB	.DAT	4K	-- " --
TTYPEXC	.DAT	4K	-- " --
TTYPEXD	.DAT	4K	-- " --
TTYPEXE	.DAT	4K	-- " --
TTIPEXF	.DAT	4K	-- " --
TTYPEXG	.DAT	4K	-- " --
TTYPEXH	.DAT	4K	-- " --
TTAPEXI	.DAT	4K	-- " --
TTYPEXJ	.DAT	4K	-- " --
XDIR	.COM	4K	ADVANCED DIR COMMAND
XDIR	.DOC	4K	DOC FOR XDIR.COM

NOTES

ATTEN.COM

This program will help you design an attennuator pad to match Z in to Z out. It questions for design type, Zin and Zout, finally drawing a circuit diagram of the finished attenuator.

AUTOLOAD.COM
AUTOLOAD.DOC

The program AUTOLOAD.COM implements the CP/M function which allows a program to be automatically loaded and run, following the cold boot. The mechanics of this are described by Kelly Smith in Microsystems, Vol. 3 (1982), #1, page 88.

BAUDSET.A
BAUDSET.D

Utility to adjust the baud rate of serial devices atached to I/O ports. System dependent

CPMFTH .ASC
FTHCPM .DOC
FTHCPM .SCR

Allows transfer of files between Forth and CP/M. Standard CP/M text files as created by ED or other compatible editors may be converted to Forth screens and vice versa. Forthcpm is written for Osborne (software works) Forth, but should be modifiable for use with other Forths.

PBH .ASM
PBH .COM

PRINT BLOCK TITLES
This program is a merging of the "prnt" file from the Cpm user group library and "Masthead" by Emilio D. Iannuccillo, based on the Basic version by Leo P. Biese. Interface Age magazine 8/80 pg 122. The two files were merged on 3-1-81 by C. B Mueller with assistance from Don Laycock.

How to use:

To print a file with no block title enter;

PBH filename.type<cr>

To print a file with a block title the same as the file name enter;

PBH filename.type<sp>*

To print a file with a desired block title enter;

PHH filename.type<sp>title

To print a block title only enter;

*<sp>title

8 characters will fit on a 8 1/2" wide page and 14 characters will fit on a 14" wide page.
To change the number of characters change maxchr in the equate tables.

RBBS-USE.DOC
RBBS31 .ASC
RBBS31A .DOC
RBSUTL31.COM

Run your own Bulletin Board software

If you want to put up a bulletin board or a program exchange system that runs under CP/M (and if you don't want to write all your own software from scratch) you have two choices: Ward Christensen's and Randy Suess's assembly-language CBBS and the basic-language RBBS written by Howard Moulton, Bruce Ratoff, Ron Fowler, Tim Nicholas and others.

RBBS
The main (and only) alternative to the CBBS series is RBBS or its two RCPM-specialized descendants, Enterbbs and Minirbbs.

RBBS is easy to install. if you have a very modest knowledge of Basic and own or have access to Bascom, you can get a RBBS system up and running in a single evening, while installing CBBS takes several days for a crack assembly-language programmer. To the RCPM operator its use of a Mbasic-type sequential message file is also an advantage access to messages can be controlled simply by declaring the messages file a $SYS file,

and can easily be protected from Xmodeming by using TAG2.COM.

==========

SCRAMBLE.COM
SCRAMBLE.DOC

Scramble is a command used to encode a CP/M file.

The format of the command is:

SCRAMBLE FILENAME.EXT PASSWORD.

where "Password" is an 8 character password made of characters perissible in a file name (i.e. no ".", etc). To obtain a good "initial seed" for the scrambling process, no character in the password may appear more than twice.

The requested file is scrambled, and re-written in place. to un-scramble the file, the identical command is issued, i.e. scramble filename.type password. This is because scramble does an "exclusive-or" type modification to the file, and doing two identical exclusive-or's to data result in the same data being retuned.

==========

SETC8SD .ASM
SETC8SD .COM

Mix an 8" drive with you normal ones.
Set drive c to 8 inch single density.
Track-to-track stepping rate of drive 6 ms.
Effective until next cold boot.

==========

SUPERSUB.ASM
SUPERSUB.COM
SUPERSUB.DOC

SUPERSUB
A replacement for the CP/M SUBMIT utility

One of the most useful programs provided with CP/M is the submit utility, which allows system commands to be read from a disk file for automatic processing. It's command-line substitution facility provides a convenient means of developing "prototype" command files, with actual parameter substitution performed when the command file is invoked.

There are, however, some shortcomings associated with submit that recur frequently. Frequently enough to inspire me to write my own batch processor, Supersub, that completely replaces, and is fully compatible with, the standard submit utility.

==========

SYNONYM3.ASM
SYNONYM3.DOC

SYNONYM VERSION 3.0
Synonym is a utility that generates a .COM program which is a loader for another .COM program. The generated program, when executed, will invoke another CP/M command. It's really not as complex as it sounds.

==========

TEMP.COM

Converts temperatures from C to F. Personally I think it should be renamed, as TEMP is often used for non-secure temporary files on a disk.

==========

TAGCPM .BAS
TRAFFIC .TXT

Train Traffic Automatically Generated Program
original trs-80 basic program by Bob Fink published in Model Railroader, February 1984

Mbasic version works with CP/M-80, CP/M-86 PCDOS and MSDOS implementations of Microsoft Basic.

==========

TTHELP0 .DAT
TTHELP1 .DAT
TTHELP2 .DAT
TTHELP3 .DAT
TTHELP4 .DAT
TTHELP5 .DAT
TTKEYBD .DAT
TTYPE .BAS
TTYPE .DOC
TTYPE .OBJ
TTYPE .WST
TTYPEXA .DAT
TTYPEXB .DAT
TTYPEXC .DAT
TTYPEXD .DAT
TTYPEXE .DAT
TTIPEXF .DAT
TTYPEX0 .DAT
TTYPEXH .DAT
TTAPEXI .DAT
TTYPEXJ .DAT

TOUCHTYPE TUTOR
TTYPE (touchtype tutor) is an application programme designed to aid the user to quickly gain touch typing skills using the computer keyboard and the computer, of course.

Some of the features are:

* Lots of help, which you can expand.
* Onscreen representation of a 'qwerty' keyboard to give visual feedback.
* Graduated exercises with prompts (or comments) throughout.

This version, (2.0), of TTYPE is written for direct cursor addressing terminals, so if yours isn't, then you may either do a complete rewrite or wait for the non-cursor addressing version.

==========

IUG-5 548K

SCAN	.COM	12K	MAGAZINE ABSTRACTS SCANNER
MAGS	.DOC	4K	-- " --
MAGS	.TXT	22K	-- " --
CIRDATA	.TXT	40K	-- " --
OTHELLO	.COM	24K	OTHELLO GAME
PRINT	.COM	4K	PRINTS A FILE TO HARDCOPY
SECRTARY	.COM	12K	WORD PROCESSOR
SECRTARY	.ASM	60K	-- " --
SECRTARY	.DOC	4K	-- " --
USERMAN0	.TXT	4K	-- " --
USERMAN1	.TXT	24K	-- " --
USERMAN2	.TXT	20K	-- " --
USERMAN3	.TXT	16K	-- " --
USERMAN4	.TXT	20K	-- " --
USERMAN5	.TXT	8K	-- " --
ADDEND2	.TXT	4K	-- " --
ADDENDUM	.TXT	8K	-- " --
DEMOFILE	.TXT	4K	-- " --
SAMPDATA	.DAT	4K	-- " --
SAMPLTTR	.TXT	4K	-- " --
MERGE	.BAS	12K	-- " --
MERGE	.COM	24K	-- " --
XDIR	.COM	4K	ADVANCED DIR COMMAND

NOTES

==========

SECRTARY	.COM	12K
SECRTARY	.ASM	60K
ADDEND2	.TXT	4K
ADDENDUM	.TXT	8K
SECRTARY	.DOC	4K
USERMAN0	.TXT	4K
USERMAN1	.TXT	24K
USERMAN2	.TXT	20K
USERMAN3	.TXT	16K
USERMAN4	.TXT	20K
USERMAN5	.TXT	8K
SAMPDATA	.DAT	4K
SAMPLTTR	.TXT	4K
MERGE	.BAS	12K
MERGE	.COM	24K

Secrtary word processor

Once upon a time many years ago when north star first started putting out systems, there was a need for a word processor that was easy to use, easy to learn, and cheap. There were very few word processors available and none available for north star dos. Video terminals and full screen editors were not common, most systems used ASR33 teletypes for I/O and most people had much less than 64k of memory.

This led to the development of Maryelln, a word processor for the north star DOS. It used less than 8k of memory so the rest of ram could hold the text, it was line oriented instead of screen oriented so it could be used with a TTY, and most of the commands were similar to the ones used in north star basic. Anyone knowing basic could immediately use maryelln. The source code was assembled using an early assembler that required that all of the code be resident in memory at one time. therefore to save memory, comments were kept to a minimum. Later it took 2, 3 and eventually 5 files to assemble separately and link together to form the final program. Maryelln did all the usual editor/word processor functions: edit, load, save, justify, line fill, center, number pages, move, copy, find, list, print, and change.

Later, maryelln, was greatly enhanced and was renamed Secrtary (8 characters) because of so many questions about why it was called Maryelln. The enhanced features included a more sophisticated installation procedure, underlining, imbedded commands, line wrap around on input, automatic insertion of names and addresses from a basic file, additional commands like status, and the ability to load north star basic programs in token format, convert them to ascii, and edit them as a normal secrtary file. I should point out that a "normal secrtary file" used the format of the old assembler or programs such as XEK, SCS etc where the line consisted of a character count, 2 byte binary line number the text, and a carriage return (no line feed).

Secrtary still consisted of five source files which had to be assembled separately and there were very few comments. Secrtary is still popular today on north star systems. the source code was merged together and maintained under CP/M however (some 5000 lines of it). Later, a CP/M version was implemented. By this time, however, full screen editors were popular and the limitation of line numbers and memory resident files was unpopular for large text files even with the "chain" command.

Secrtary is now available to CPM users for people who need a word processor for a few letters but do not want the expense or complications of the more extensive full screen editors.

To get secrtary to work, first assemble it using ASM or MAC. it uses 8080 code not z80 code. The full user manual is contained in files USERMAN0.TXT, USERMAN1.TXT, USERMAN2.TXT etc. After loading the program, execute it and enter the command "LOAD USERMAN1.TXT". The program will load the file and tell you the size. Then type "LIST" (no quotes) and the file will be listed on the screen. To get a hard copy of it, first enter the command "DEVICE L" to have the listing or printing go to the list device. (List shows line numbers and control commands, print justifies, and applies the commands to the final text). After setting the device to the list device, simply enter the command print. Each file will "chain" in the next file and you will have the full user manual which will teach you to use the other features. "DEVICE C" returns the print or listing to the console device.

==========

SCAN	.COM	12K
MAGS	.DOC	4K
MAGS	.TXT	22K
CIRDATA	.TXT	40K

INDEX TO SOME PERSONAL COMPUTING MAGAZINES 1978-83

MAGS is an ASCII file containing 3170 one-line abstracts of articles appearing in four personal computing magazines.

SCAN.COM is simple program which asks the user for one or more keywords, reads MAGS from beginning to end, and writes a file of only those lines containing the keywords. The output file may then be printed, or if large, be SCANned itself for further keywords. The abstracts will only rarely answer a query by themselves - they are pointers to where information may be found.

The structure of each entry consists of a substitute title, optionally further keywords by which the article might be sought, the name (usually abbreviated) of the journal, the date of the issue, and the page number of the article. All major articles in each issue have been abstracted with the exception of games and fiction. The entries are arranged in page number order within each issue,in the magazine order shown below within each month, and in chronological order from the most modern to the most ancient. The coverage is shown below.

BYTE	USA	Byte	3/80 to 12/83
PCW	UK	Personal Computer World	5/78 1st to 12/83
PRAC	UK	Practical Computing	9/78 1st to 12/83
SOFT	UK	Soft	6/83 1st to 12/83

The currency sign ($) means a pound Sterling if the journal is British, a US dollar if American.

The following abbreviations have been used.

AI - Artificial Intelligence
CAI - Computer Assisted Instruction
COMPTOCOMP - Computer to Computer Communication
CS - Computer Science
DBMS - Database Management System
I/O - Input/Output
LAN - Local Area Network
LCD - Liquid Crystal Display
MC - Machine Code
OS - Operating System
PC - Personal Computer
SBC - Single Board Computer

To use the index type SCAN MAGS FILENAME, where FILENAME is the user-supplied name of the output file.

Upon receiving the ENTER TITLE ptompt, you may type any identifying description of the search. What you enter will simply appear as the first line of the output file. The program then asks for the first keyword you are searching for with the prompt 'Enter target (1)'. Enter the exact upper-case characters you are looking for, e.g. IBM PC. The press return. You may then enter further keywords in the same way. Terminate the process by pressing RETURN with no entry. When the program has finished, it will end with an END OF FILE message. You may then print or type the output file.

IUG-6 540K

ANTONYMS.COM	20K	GAME
BACCRRT .COM	16K	GAME
BANNER .COM	16K	DRAWS LARGE LETTERS
BASEBALL.COM	24K	GAME
BIRTHDAY.COM	20K	GAME
CHASE .COM	16K	GAME
CHESS .COM	24K	GAME
CIVILWAR.COM	20K	GAME
CLIMATES.COM	16K	GAME
CRAPS .COM	16K	GAME
CRAZY8 .COM	20K	GAME
DOTS .COM	24K	GAME
DRAGRACE.COM	16K	GAME
FOOTBALL.COM	20K	GAME
GREEKRTS.COM	16K	GAME
HIDESEEK.COM	16K	GAME
IQUEEN .COM	16K	GAME
KING .COM	20K	GAME
MASTERMD.COM	16K	GAME
MEMBRAIN.COM	16K	GAME
OTHELLO .COM	24K	GAME
RADIX .COM	16K	NUMBER BASE CONVERSION
ROULETTE.COM	16K	GAME
SHOP .COM	20K	GAME
SNOOPY .COM	16K	GAME
SQUARE .COM	20K	GAME
SWARMS .COM	20K	GAME
SYNONYMS.COM	20K	GAME
XDIR .COM	4K	ADVANCED DIR COMMAND

NOTES

ANTONYMS.COM

In this program you will find a collection of antonyms. A antonym of a word, in this program, will mean another word in the English language which has the opposite or very nearly the opposite meaning.

The computer chooses a word and you must supply the antonym or else type help, in which case the computer will tell you the answer.

BACCRRT .COM

Plays baccarat simulating the full casino rules. Full instructions in the game.

BANNER .COM

Draws a message as a banner headline. Sizes 1 through 7 with two different layouts.

BASEBALL.COM

You run a baseball team against the ccmputers baseball team. Full instructions are in the game.

BIRTHDAY.COM

You give your name, age and birthday. The computer prints out your age in days and gives a breakdown of the planitary structure at that time with a personality analysis.

CHASE .COM

You are in a high voltage maze with security machines trying to destroy you. You must try to survive.

CIVILWAR.COM

Civil war simulation. You must try to win it. Your responses could change history.

CRAPS .COM

Craps game. Casino simulation with full instructions in the program.

CRAZY8 .COM

This is a card game and has full instructions in the program.

DOTS .COM

You play against the computer to link dots to win. Full instructions in the program.

DRAGRACE.COM

You design, build and drive a dragster. Either against a friend or the computer. Full instructions in the programs.

FOOTBALL.COM

You are the coordinator of a football team and you must try to win. Full instructions in the program.

GREEKRTS.COM

Question and answer game on greek roots.

HIDESEEK.COM

You are to find the four hidden players in a ten by ten grid.

IQUEEN .COM

You and the computer use one queen. Chess moves only, taking turns to move her, and the first to get her to the lower right hand square wins.

KING .COM

Dunno about this. It hung up my machine when I tried it. But i don't use a VDU2K on which it will probally be ok.

MASTERMD.COM

Usual five numbers guess with the computer giving you correct and incorrect guesses.

MEMBRAIN.COM

Role play. You are a cell membrain. You are in charge of various body functions which you try to maintain.

OTHELLO .COM

Play othello against the computer. Full instructions inside the program.

RADIX .COM

Changes the base of a given number to another base. Range of bases is 2-36.

ROULETTE.COM

Game of roulette. Full instructions inside the program.

SHOP .COM

You can do your shopping via the computer. You input a list of things you buy and the computer will present you with a bill.

SNOOPY .COM

Draws a picture of snoopy.

SWARMS .COM

A swarm of bees attack you. Usual game plan. Full instructions in the program.

SYNONYMS.COM

The computer chooses a word and you must supply another word with the same meaning.

Yawn.....

IUG-7 536K

```
RETRIEVE.CMD   4K  DATA REFERENCES LOCATOR
RETRIEVE.DOC  52K  ... NEEDS DBASE2
AUTHORS .NDX   8K          -- " --
AUTONUMB.CMD   4K          -- " --
ABKDB2  .NDX   4K          -- " --
ACTION  .REF   8K          -- " --
EDIT    .BK    4K          -- " --
EDIT    .CMD   8K          -- " --
FASTFIND.CMD   8K          -- " --
FIND    .CMD   8K          -- " --
INDEX   .CMD   4K          -- " --
KWORDS  .NDX   8K          -- " --
LABEL   .CMD   4K          -- " --
NEWENTRY.CMD   4K          -- " --
SOURCE  .NDX   4K          -- " --
REFS    .DBF  20K          -- " --
C       .CMD   8K  INTIGRATED CIRCUIT LOCATOR
ADD     .CMD   4K  ... NEEDS DBASE2
CHANGE  .CMD   4K          -- " --
DELETE  .CMD   4K          -- " --
TYPE    .CMD   4K          -- " --
NUMBER  .CMD   8K          -- " --
TYP     .NDX   8K          -- " --
NUM     .NDX   4K          -- " --
CHIPS   .DBF  12K          -- " --
BOOKS   .CMD  12K  DATA MANAGEMENT PROGRAM FOR
BOOKS   .DOC   4K  ... BIBLIOGRAPHIC DATA.
ADD     .BK    8K  ... NEEDS DBASE2
DUPCHECK.BK    4K          -- " --
DUPREC  .BK    4K          -- " --
INIT    .BK    4K          -- " --
MAIN    .BK    8K          -- " --
MAINTAIN.BK   12K          -- " --
PRINT   .BK    4K          -- " --
PURGE   .BK    4K          -- " --
REPORT  .BK    8K          -- " --
SEARCH  .BK    8K          -- " --
SEARCH1 .BK    8K          -- " --
SPECIAL .BK   20K          -- " --
TITLE   .BK    8K          -- " --
VERIFDEL.BK    4K          -- " --
VERIFNEW.BK    4K          -- " --
WSFILE  .BK   12K          -- " --
TREF1   .NDX   4K          -- " --
TBKDB2  .NDX   4K          -- " --
BKDB2   .DBF   4K          -- " --
REFERENC.CMD  12K  DATA MANAGEMENT PROGRAM  FOR
REFERENC.DOC   4K  ... REFERENCE DATA.
PURGE   .REF   4K  ... NEEDS DBASE2.
PRINT   .REF   4K          -- " --
MAINTAIN.REF  12K          -- " --
MAIN    .REF   8K          -- " --
DUPREC  .REF   4K          -- " --
DUPCHECK.REF   4K          -- " --
COD     .REF   8K          -- " --
ADD     .REF   8K          -- " --
INIT    .REF   4K          -- " --
LOOSLEAF.REF   4K          -- " --
WSFILE  .REF  16K          -- " --
VERIFNEW.REF   4K          -- " --
VERIFDEL.REF   4K          -- " --
TITLE   .REF   8K          -- " --
SPECIAL .REF  16K          -- " --
SEARCH1 .REF   4K          -- " --
SEARCH  .REF   8K          -- " --
REPORT  .REF   8K          -- " --
EDIT    .REF   4K          -- " --
AREF1   .NDX   4K          -- " --
CREF1   .NDX   4K          -- " --
REF1    .DBF   4K          -- " --
STAT    .COM   8K  CP/M STATUS COMMAND FILE.
XDIR    .COM   4K  ADVANCED DIR FUNCTION.
```

NOTES

```
RETRIEVE.CMD
RETRIEVE.DOC  Read this before using RETRIEVE.
REFS    .DBF
RETRIEVE.CMD
NEWENTRY.CMD
AUTONUMB.CMD
FIND    .CMD
FASTFIND.CMD
INDEX   .CMD
LABEL   .CMD
EDIT    .CMD
REFS    .DBF
AUTHORS .NDX
SOURCE  .NDX
KWORDS  .NDX
```

Retrieve is a collection of DBASE2 command files which allow creation of a reference article citation database for retrieval by an assigned reference number, author, title, source, or key word. To use the system, you must possess a version of the dbase2 assembly-language relational database management system.
The retrieve system allows the user to store reference article citations by authors, title, source, key words and a code to be used in filing the original article.

```
C       .CMD
ADD     .CMD
CHANGE  .CMD
DELETE  .CMD
NUMBER  .CMD
TYPE    .CMD
NUM     .NDX
TYP     .NDX
CHIPS   .DBF
```

"CHIPS" is a DBASE2 program to help locate intigrated circuit chips in a bin storage system.

****
* Listings printed in iugn-16 *
****

A programme to access the "CHIPS" database
Search for IC by number
Search for IC by type
Add IC's to database
Delete IC's from database
Change stored IC data
Leave program

It was developed for the VDU2K card and later modified it to suit the televideo 950 vdu. The program will cater for both screen widths so it can be run by any interak user. There are plenty of comments so that it is easy to see how it works!

At the moment it is only for IC's, but could be modified to include all other semiconductor stocks if you have the time to look up all the data.

The program is modular so that with a few changes of headings and variables it could easily be modified to suit other uses.

This program was configured to run on a 64 column VDU2K card or a serial vdu with 80 columns. This listing is therefore written for the minimum width. Use is made of the VDU2K's inverse video facility, (variables rev and norm), to highlight certain characters. This does not affect the programmers televideo vdu, but it should be born in mind if a different type is used.
The program is menu driven and,(hopefully),idiot proof!.

Use "DBASE.C" to initiate the program.

```
BOOKS   .CMD
BOOKS   .DOC
DUPCHECK.BK
DUPREC  .BK
INIT    .BK
MAIN    .BK
MAINTAIN.BK
PRINT   .BK
PURGE   .BK
REPORT  .BK
SEARCH  .BK
SEARCH1 .BK
SPECIAL .BK
TITLE   .BK
VERIFDEL.BK
VERIFNEW.BK
WSFILE  .BK
TREF1   .NDX
TBKDB2  .NDX
CREF1   .NDX
AREF1   .NDX
BKDB2   .DBF
```

Data management program for bibliographic data.

Type 'DBASE BOOKS' and the initialization program will lead you through what you have to do. If you just want to check out the program, leave DBASE in drive A and the all of the files in drive B. From drive A type DBASE. At the dot prompt type 'SET DEFAULT TO B'. At the next dot prompt type 'DO BOOKS'.

==========

```
REFERENC.CMD
REFERENC.DOC
ADD     .REF
COD     .REF
DUPCHECK.REF
DUPREC  .REF
INIT    .REF
LOOSLEAF.REF
MAIN    .REF
MAINTAIN.REF
PRINT   .REF
PURGE   .REF
REPORT  .REF
SEARCH  .REF
SEARCH1 .REF
SPECIAL .REF
TITLE   .REF
VERIFDEL.REF
VERIFNEW.REF
WSFILE  .REF
REF1    .DBF
```

Data management program for reference data.

This program is self explanatory and self documented. Type 'DBASE REFERENCE' and the initialization program will lead you through what you have to do.
If you just want to check out the program, leave dbase in drive A and the all of the files in drive B. From drive A type DBASE. At the dot prompt type 'SET DEFAULT TO B'. At the next dot prompt type 'DO REFERENCE. (Note a file name can only be eight letters long when you type 'DBASE REFERENCE' the computer ignores the last 'E'. In the DBASE program, you must omit the last 'E'.

==========

Tom Evans will be moving location in the very near future, could ALL mail and Membership applications be temporarily sent c/o GREENBANK till further notice.

Ta muchly

Tom.

MISSING

Registered in England 1970
Company Reg No 1507905
VAT Reg No 165 5021 81
Telephone No 051-645 3391

Greenbank Electronics
460 New Chester Road
Rock Ferry, Birkenhead
Merseyside. L42 2AE

OUR REF: LO/IUGN/16
DATE: January 1988

To all members of
Interaktion User Group
(This letter distributed
with Newsletter Number 16)

Dear Member

NB This is not part of the IUG Newsletter (IUG is editorially independent of Greenbank Electronics), but is being sent to you directly from Greenbank.

To us again has fallen the lot of printing and distributing the IUG newsletters.

We are sorry for the long interval bewteen this and the last, but please don't blame Tom or Bob or Charlie; it's not their fault. Don't blame us either; we print and distribute this newsletter on behalf of the group, entirely without profit, in fact at a total loss at the moment, since nobody including ourselves has yet had a penny for expenses. In round terms the income from 16 issues of a newsletter for 400 members paying at the rate of 7.00 per 4 issues should be 16 x 400 x 7 / 4 = 11,200 pounds. In fact the amount received in subscriptions to date is little more than 1,500 pounds. Thus the majority, the members who have not paid their subscriptions, have nothing to complain about, but our apologies must go to those who do pay regularly; they like the rest of us are being taken for a bit of a ride.

Bob and Tom are too nice to say it, but I think it is worthwhile making a few points on their behalf:

1. The membership fee entitles the member to the use of the Interaktion disk library (no waiting involved - Charlie at present is managing a "by-return" service).

2. The membership fee entitles the member to 4 issues of the newsletter, as and when it is published, nominally once every three months, but if only one issue came out per year then their membership fee would last four years. Not entirely satisfactory, but the only fair way to do it in the circumstances.

3. The cover date on the newsletter simply is a guide to the date the information was produced; this obviously can be widely different from the date it is printed and posted as there is no official printer for the group. (We at Greenbank Electronics help out at the moment, but you must appreciate that this printing has to fit in with our own work load, and depends on what spare capacity we have for printing, collating, distribution, and so on.)

Yours sincerely,

David M. Parkins

David M Parkins, B Eng (Hons),
Partner, Greenbank Electronics

Will any members normally paying by standing order to the Staffordshire bank account please note that this is now closed.

For the time being only cheques and postal orders can be accepted, made payable to "Interaktion" and sent to The Treasurer IUG, c/o Greenbank Electronics.

January 1988